mémento

CLEFS POUR LE ZX 81 ET TIMEX 1000

Jean-François Séhan

Editions du PSI

CLEFS POUR LE ZX 81
ET TIMEX 1000

par
Jean-François Séhan

Editions du P.S.I.
1983

La collection « **MEMENTOS** » est constituée de recueils de données sur un type de matériels. Huit titres sont actuellement disponibles dans cette collection :

— Clefs pour le TRS-80 — tomes 1 et 2 — Rémy Pineau
— Le Basic de A à Z — Jacques Boisgontier
— Clefs pour l'Apple II — Nicole Bréaud-Pouliquen
— CP/M mot par mot — Yvon Dargery
— Clefs pour le ZX-81 — Jean-François Séhan
— Clefs pour le CBM — Daniel-Jean David
— Clefs pour le Vic — Daniel-Jean David

D'autres ouvrages relatifs au ZX-81 :

— La pratique du ZX-81 — tome 1 — Xavier Linant de Bellefonds
Collection « MATERIELS »
série bleue
— La pratique du ZX-81 — tome 2 — Marcel Henrot
Collection « MATERIELS »
série rouge
— Etudes pour ZX-81 — tomes 1 et 2 — Jean-François Séhan
Collection « PROGRAMMES »
tome 1 — série bleue
tome 2 — série rouge

RAPPELS

Les séries :

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

Série VERTE : ouvrage d'initiation ne nécessitant que des connaissances de base.
Série BLEUE : suppose une connaissance élémentaire du sujet traité.
Série ROUGE : ouvrage d'approfondissement, niveau de complexité moyen.
Série NOIRE : ouvrage d'approfondissement, niveau de complexité élevé.

Les collections :

Les ouvrages d'Edition du PSI, actuellement au nombre de 100, sont répartis en collections :

« LANGAGES », « MATERIELS », « PROGRAMMES », « GUIDES PRATIQUES », « MEMENTOS », « UTILISATIONS DE L'ORDINATEUR » et pour l'initiation, outre quelques livres hors collection, **« ... POUR TOUS »**.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© *Editions du P.S.I. Parc industriel nord, Bâtiment 9, 77200 Torcy Marne-la-Vallée 1983*
ISBN : 2-86595-086-7

CLEFS POUR LE ZX 81

ET TIMEX 1000

Du même auteur :

- Variations pour PC-1211
- Etudes pour ZX-81 - tomes 1 et 2
- Suites pour PC-1500
- Le TI-99 4/A à l'affiche
- Le VIC à l'affiche

PRESENTATION

Ce livre est destiné aux nombreux utilisateurs de ZX81 perdus parmi la vaste documentation mise à leur disposition. Pas de parlote inutile, mais des renseignements, et encore des renseignements !

Ce mémento permet d'accéder rapidement aux données techniques dont vous aurez besoin. Vous y trouverez la liste des instructions Basic commentées, les mnémoniques de l'assembleur Z80 et leurs codes objets, les points d'entrées variables système et les caractéristiques des principales extensions, ainsi que les codes clavier et écran, les codes erreur et le brochage des circuits intégrés.

Une liste d'astuces pour mieux utiliser l'écran, les cassettes et les programmes en langage machine complète cet ouvrage.

Et, par souci d'efficacité un index, placé en fin d'ouvrage, vous indique où trouver telle instruction Basic ou telle variable système.

SOMMAIRE

	pages
COMMANDES	9
Fonctions Basic	9
Commandes Basic	11
Opérateur logiques	14
Liste des mots-clefs par ordre de code	15
Extension haute-résolution Quicksilva	16
Extension générateur de caractères Quicksilva	17
CODES ERREUR	19
Liste des codes erreur	19
Liste des erreurs par mot-clef	21
ECRAN CLAVIER	23
Jeu de caractères	23
Codes clavier	27
Grille d'écran	29
LANGAGE MACHINE	31
Organisation interne du Z80	31
Registres du Z80	32
Schéma simplifié du ZX81	33
Jeu d'instructions du Z80	34
Codes des instructions Z80 par ordre alphabétique	40
Tableaux de désassemblage	55
CONNECTEURS	59
Connecteur arrière du ZX81	59
Brochage des circuits intégrés	60

ADRESSES

Carte mémoire 65
Adresses de la ROM BASIC (de 1000 à 1DFF) 67
Variables système (de 4000 à 407C) 75
Carte sonore QS (7FFE et 7FFF) 79

TRUCS ET ASTUCES 81

Ecran de 24 lignes 81
Affichage direct écran 81
Exécution automatique après chargement 81
Chaînage de programmes 82
Sauvegarde rapide 82
Instruction VAL 83
Langage machine dans une variable 84
Langage machine en haut de mémoire 84
Gain de place en mémoire centrale 84
Horloge temps réel 84
Caractères géants 85
Haute-résolution sur l'imprimante 85

ANNEXES 87

Conversion binaire 87
Conversion décimal/hexadécimal 88

INDEX 91

Fonctions mathématiques

Mot-clé	*Définition, exemples*
ABS	Donne la valeur absolue de l'argument : ABS -5 ⟶ 5
ACS	Donne l'arccosinus de l'argument (en radian) : ACS 0.5 ⟶ 1.0471976
ASN	Donne l'arcsinus de l'argument (en radian) : ASN 0.6 ⟶ 0.64350111
ATN	Donne l'arctangente de l'argument (en radian) : ATN 0.7 ⟶ 0.61072597
COS	Donne le cosinus de l'argument (en radian) : COS 1.6 ⟶ 0.029199522
EXP	Fonction exponentielle : EXP 10 ⟶ 22026.466
INT	Donne la valeur entière de l'argument : INT 2.5 ⟶ 2 INT -5.4 ⟶ -6
LN	Fonction logarithme naturel (base e) : LN 5 ⟶ 1.6094379
PI	Constante de valeur 3.141592653. Seuls les huit premiers chiffres sont affichés à l'écran (3.1415927).
SGN	Signe de l'argument. -1 si négatif, 0 si nul et 1 si positif.
SIN	Donne le sinus de l'argument (en radian) : SIN 1 ⟶ 0.84147099
SQR	Donne la racine carrée de l'argument : SQR 25 ⟶ 5
TAN	Donne la tangente de l'argument (en radian) : TAN 1.5 ⟶ 14.10142

Fonctions chaînes de caractères

Mot-clé	*Définition, exemples*
CHR$	Donne le caractère de l'argument (numérique) : CHR$ 38 ⟶ "A"
CODE	Donne le code du premier caractère d'une chaîne, ou 0 si la chaîne est vide : 10 A$ = "ZX-81" 20 PRINT CODE A$ donne 63 (code de Z)
LEN	Longueur d'une chaîne : 10 A$ = "ZX-81" 20 PRINT LEN A$ ⟶ 5
STR$	Convertit un nombre en chaîne de caractères : STR$ 12.5 ⟶ "12.5"
TO	Extrait une sous-chaîne d'une chaîne de caractères : A$ = "SINCLAIR" A$ (TO 4) ⟶ "SINC" A$ (3 TO 5) ⟶ "NCL" A$ (5 to) ⟶ "LAIR"
VAL	Donne la valeur numérique d'une chaîne : VAL "12.5" ⟶ 12.5

Fonctions langage machine

Mot-clé	*Définition, exemples*
PEEK	Restitue l'octet contenu dans la mémoire spécifiée : 10 PRINT PEEK 16398 20 A=PEEK 16398+256*PEEK 16399
POKE	Place dans l'octet désigné le nombre spécifié : 10 POKE 16442,4
URS	Appelle un sous-programme écrit en langage machine dans la mémoire vive ou dans la mémoire morte du Basic: 10 PRINT USR 16514 Au retour du sous-programme en langage machine, USR donne la valeur de la paire de registres BC du Z80

D : instruction utilisable uniquement en mode direct.
P : instruction utilisable uniquement dans un programme.

	Mot-clé	*Définition, exemples*
	CLEAR	Supprime les variables déjà déclarées et libère l'espace qu'elles occupaient.
	CLS	Efface l'écran en plaçant dans le fichier d'affichage 25 newlines.
D	**CONT**	Continue l'exécution du programme après une instruction STOP ou un BREAK. Cette commande effectue d'abord une instruction CLS.
	COPY	Recopie le contenu de l'écran sur l'imprimante si elle est connectée.
P	**DIM**	Dimensionne un tableau (fixe le nombre et les valeurs maximum des indices). Pour les chaînes, le dernier indice indique la longueur des chaînes du tableau : 10 A(50) 20 A(20,10) 30 A$(30,15)
D	**EDIT**	Place la ligne pointée par le curseur en mode édition.
	FAST	Place le ZX-81 en mode rapide. L'écran n'apparaît qu'à la fin du programme, pendant l'exécution d'une instruction PAUSE ou dans la saisie de données par INPUT.
P	**FOR TO STEP**	Introduit une boucle : toutes les instructions, comprises entre FOR I=A TO B STEP C et le NEXT I correspondant, seront répétées pour toutes les valeurs de I allant de A à B, de C en C. Si C n'est pas spécifiée, elle prend, par défaut, la valeur 1.
	GOSUB	Appelle le sous-programme de la ligne spécifiée : 100 GOSUB 1000
	GOTO	Effectue un saut à la ligne spécifiée : 100 GOTO 150
	IF THEN	Test logique de la forme IF condition THEN instruction : 10 IF A=10 THEN GOTO 100 100 IF A$="OUI" THEN PRINT "FAUX" 200 IF A<0 THEN LET A=1

	Mot-clé	*Définition, exemples*
P	**INKEY$**	Donne le caractère tapé au clavier ou une chaîne vide si aucune touche n'est enfoncée.
P	**INPUT**	Place dans la variable spécifiée la valeur ou les caractères tapés au clavier : 10 INPUT A 20 INPUT A$
	LET	Affecte à une variable le résultat de l'expression à droite du égal : 10 LET A=10 20 LET B=5+(2*3)
D	**LIST**	Liste le programme sur l'écran à partir du début ou de la ligne spécifiée : LIST LIST 100
D	**LLIST**	Liste le programme sur l'imprimante si elle est connectée.
	LOAD	Charge un programme du magnétophone. Si aucun nom n'est indiqué, LOAD charge le premier programme de la bande : LOAD "PROGRAMME" LOAD ""
	LPRINT	Indentique à PRINT, mais sur l'imprimante.
D	**NEW**	Efface de la mémoire vive le programme et les variables.
P	**NEXT**	Incrément de la variable correspondant au FOR (voir FOR TO STEP).
	PAUSE	Arrête l'exécution du programme en cours, pendant n trames de la télévision. N est compris entre 1 et 32767. Si N est supérieur à 32767, le programme attend qu'une touche soit enfoncée. Cette instruction restitue l'écran si on est en mode FAST. 10 PAUSE 1000 250 PAUSE 40000
	PLOT	Affiche à l'écran l'élément C,L (où C est le numéro de colonne et L la ligne ; voir chapitre Caractères) : 10 PLOT 12,24
	PRINT	Affiche à l'écran des valeurs numériques ou des chaînes de caractères : PRINT A,A$; "ZX-81"

	Mot-clé	*Définition, exemples*
	PRINT AT	Identique à PRINT, mais à l'adresse X,Y désignée : PRINT AT 10,10 ; "ZX"
	PRINT TAB	Identique à PRINT, mais en partant de la colonne désignée : 10 PRINT TAB 10 ; "BONJOUR"
	RAND	Fixe la séquence des nombres pseudo-aléatoires à partir de N, N étant compris entre 0 et 65536 : 10 RAND 2000 20 PRINT RND
P	**REM**	Débute une ligne de commentaires dans un programme.
P	**RETURN**	Retour au programme principal, après l'appel d'un sous-programme par GOSUB.
	RND	Donne un nombre pseudo-aléatoire : 10 PRINT RND 20 A=RND*10
	RUN	Exécute le programme en mémoire, à partir de la première ligne ou du numéro de ligne spécifiée : RUN RUN 100 Cette commande effectue d'abord un CLS.
D	**RUBOUT**	Supprime le dernier caractère tapé au clavier.
	SAVE	Sauve sur le magnétophone le programme en mémoire : SAVE "PROGRAMME"
	SCROLL	Fait remonter l'écran d'une ligne en insérant une ligne de blancs en bas de l'écran.
	SLOW	Place le ZX-81 en mode lent. Le fichier d'affichage reste en permanence sur l'écran.
P	**STOP**	Arrête l'exécution du programme en cours. On peut alors visualiser des variables et repartir par CONT.
	UNPLOT	Efface de l'écran l'élément C,L (où C est le numéro de la colonne, et L la ligne ; voir chapitre Caractères). 10 UNPLOT 12,24

OPERATEURS LOGIQUES

Priorité	*Mot-clé*	*Définition, exemples*
3	**AND**	Fonction ET : - numérique : A AND B = A si B≠0 0 si B=0 - chaîne : A$ AND B = A$ si B≠0 "" si B=0
4	**NOT**	Fonction NON : NOT A = 0 si A≠0 1 si A=0
2	**OR**	Fonction OU : A OR B = 1 si B≠0 A si B=0
6	+	Additionne des nombres ou concatène des chaînes de caractères.
6	-	Soustraction de valeurs numériques.
8	*	Multiplie des valeurs numériques.
8	/	Divise des valeurs numériques.
10	**	Elévation à la puissance.
		Pour les six comparaisons qui suivent, les opérandes doivent être du même type (numériques ou chaînes de caractères). Le résultat est 1 si la comparaison est vraie, 0 si elle est fausse.
5	=	Egal.
5	<	Inférieur.
5	>	Supérieur.
5	< =	Inférieur ou égal.
5	> =	Supérieur ou égal.
5	<>	Différent.

LISTE DES MOTS-CLEFS PAR ORDRE DE CODE

Dec.	*Hexa.*	*Mot-clé*	*Dec.*	*Hexa.*	*Mot-clé*
18	12	>	220	DD	>=
19	13	<	221	DE	<>
20	14	=	222	DF	**THEN**
21	15	+	223	E0	**TO**
22	16	-	224	E1	**STEP**
23	17	*	225	E2	**LPRINT**
24	18	/	226	E3	**LLIST**
64	40	**RND**	227	E4	**STOP**
65	41	**INKEY$**	228	E5	**SLOW**
66	42	**PI**	229	E6	**FAST**
193	C1	**AT**	230	E7	**NEW**
194	C2	**TAB**	231	E8	**SCROLL**
196	C4	**CODE**	232	E9	**CONT**
197	C5	**VAL**	233	EA	**DIM**
198	C6	**LEN**	234	EB	**REM**
199	C7	**SIN**	235	EC	**FOR**
200	C8	**COS**	236	ED	**GOTO**
201	C9	**TAN**	237	EE	**GOSUB**
202	CA	**ASN**	238	EF	**INPUT**
203	CB	**ACS**	239	F0	**LOAD**
204	CC	**ATN**	240	F1	**LIST**
205	CD	**LN**	241	F2	**LET**
206	CE	**EXP**	242	F3	**PAUSE**
207	CF	**INT**	243	F4	**NEXT**
208	D0	**SQR**	244	F5	**POKE**
209	D1	**SGN**	245	F6	**PRINT**
210	D2	**ABS**	246	F7	**PLOT**
211	D3	**PEEK**	247	F8	**RUN**
212	D4	**USR**	248	F9	**SAVE**
213	D5	**STR$**	249	FA	**RAND**
214	D6	**CHR$**	250	FB	**IF**
215	D7	**NOT**	251	FC	**CLS**
216	D8	**	252	FD	**UNPLOT**
217	D9	**OR**	253	FE	**CLEAR**
218	DA	**AND**	254	FF	**RETURN**
219	DB	< =	255		**COPY**

Abréviations	*Mot-clé*	*Définitions*
BL	**BLACK**	Tous les tracés deviennent noirs sur fond blanc.
BO	**BOX X,Y**	Trace un rectangle des coordonnées actuelles aux coordonnées X,Y.
CL	**CLEAR**	Efface l'écran haute-résolution de la couleur inverse à celle définie par BLACK ou WHITE.
CO	**COPY**	Recopie le contenu de l'écran sur l'imprimante si elle est connectée.
DO	**DOWN**	Déplace le contenu de l'écran d'une rangée de pixels vers le bas.
DR	**DRAW X,Y**	Trace une ligne de la position du curseur aux coordonnées X,Y.
LE	**LEFT**	Déplace le contenu de l'écran d'une colonne de pixels vers la gauche.
MO	**MOVE X,Y**	Déplace le curseur aux coordonnées X,Y.
PL	**PLOT X,Y**	Affiche un pixel aux coordonnées X,Y et dans la couleur en cours.
PR	**PRINT A$**	Affiche à l écran le contenu de la variable A$ (uniquement) en partant de la position courante.
RI	**RIGHT**	Déplace le contenu de l'écran d'une colonne de pixels vers la droite.
SC	**SCROLL**	Déplace le contenu de l'écran vers le haut pour laisser une fenêtre de 3 lignes en bas de l'écran pour les messages utilisateurs.
	UP	Déplace le contenu de l'écran d'une rangée de pixels vers le haut.
TH	**WITH**	Tous les tracés deviennent blancs sur fond noir.

Remarque : toutes les commandes sont appelées à partir du Basic par la fonction LET L=USR 10240, suivie d'une ligne REM contenant la liste des commandes à effectuer :

```
10 LET L = USR 10240
20 REM BLACK CLEAR
```

Adresse	*Mot-clé*	*Définition, exemples*
16581	**CLR**	Efface la mémoire vive contenant les caractères programmables. A l'initialisation ces caractères ont des formes aléatoires : 10 LET CLR=16581 20 LET L=USR CLR
16728	**QCOPY**	Copie sur l'imprimante le contenu de l'écran, comme la commande COPY du Basic, mais en tenant compte des caractères programmés : 10 LET QCOPY=16728 20 LET L=USR QCOPY
16619	**QCOPY**	Charge en mémoire vive les caractères dont la définition est placée dans des REM : 10 LET QLOAD=16619 20 LET L=USR QCOPY 30 REM A0103070F1F3F7FFF' 40 REM B ... etc... 51 REM STOP La liste des caractères doit toujours être suivie d'un REM STOP (shift A). Le numéro de cette ligne doit toujours être impair.
16689	**QPRINT**	Imprime les caractères contenus dans l'instruction REM qui suit : 10 LET QPRINT=16689 20 LET L=USR QPRINT 30 REM AABAACAABAA
16601	**SINC**	Recopie les caractères de la ROM Sinclair dans la mémoire vive de la carte génératrice de caractères : 10 LET SINC=16601 20 LET L=USR SINC

Remarque : les sous-programmes en langage machine sont chargés à partir du programme "CHRS" vendu avec la carte. Ce programme se place entre les lignes 1 et 14.

LISTE DES CODES ERREUR

Code	*Signification*	*Cas de :*
0	Le programme s'arrête normalement après l'exécution de la dernière ligne. Aucune erreur n'a été rencontrée.	
1	NEXT sans FOR. La variable de contrôle d'un NEXT n'a jamais été définie par FOR, mais existe cependant dans le programme, comme variable normale.	**NEXT**
2	Variable indéfinie. La variable utilisée n'a pas été définie préalablement, soit par une instruction LET ou INPUT, soit par l'instruction DIM s'il s'agit d'une variable indicée.	
3	Indice hors intervalle. La valeur de l'indice ou le nombre d'indices d'un tableau dépasse la taille définie par l'instruction DIM.	
4	Mémoire remplie. Toute la mémoire vive du ZX-81 est prise par les lignes de programme, les variables et le fichier d'affichage. En version de base (1k), ceci peut se produire aussi par les instructions PRINT, PLOT, UNPLOT ou LIST.	**LET, INPUT, DIM, FOR GOSUB.** en 1ko **PRINT, PLOT UNPLOT, LIST**
5	Ecran plein. L'instruction CONT dégage de la place en effaçant l'écran et continue le programme. Une instruction SCROLL dans le programme ajoute une ligne en bas de l'écran et évite l'erreur 5.	
6	Nombre trop grand. Le résultat d'une opération donne un nombre supérieur à 10 puissance 38.	
7	RETURN sans GOSUB. Le programme a rencontré une instruction RETURN sans qu'il y aie jamais eu de GOSUB.	**RETURN**

LISTE DES CODES ERREUR

Code	*Signification*	*Cas de :*
8	Vous avez utilisé l'instruction INPUT comme commande directe.	**INPUT**
9	Point d'arrêt. Le programme a rencontré une instruction STOP. Le programme peut continuer en utilisant CONT, mais ne tente pas l'exécution d'une nouvelle instruction STOP.	**STOP**
A	Argument erroné. Une fonction mathématique a pour argument une chaîne de caractères.	**SQR, LN, ASN ...**
B	Entier hors intervalle. L'argument requis est supérieur à 65535 ou négatif.	**RUN, GOTO, DIM, USR...**
C	L'argument de VAL (chaîne) n'est pas une expression numérique autorisée.	**VAL**
D	Arrêt par BREAK. Le programme a été interrompu par la touche BREAK. Le même compte-rendu apparaît après une fonction cassette ou imprimante.	
F	Commande SAVE erronée. L'argument de SAVE est une chaîne vide ou de plus de 10 caractères.	

Clé	*Codes erreur*	*Clé*	*Codes erreur*
*	6	LIST	4,5,B
**	6	LLIST	B,D
+	6	LN	A
-		LOAD	4,D
/		LPRINT	D
<		NEW	
< =		NEXT	1,2
< >		NOT	
=		OR	
>		PAUSE	B
> =		PEEK	B
ABS		PI	
ACS	6,A	PLOT	4,B
AND		POKE	B
ASN	6,A	PRINT	3,4,5
ATN		RAND	B
AT	4,B	REM	
CHR$	B	RETURN	7
CLEAR		RND	
CLS		RUN	B
CODE		SAVE	D,F
CONT	B	SCROLL	
COPY	D	SGN	
COS		SIN	
DIM	4,B	SLOW	
EXP	6	SQR	A,B
FAST		STEP	
FOR	4	STOP	9
GOSUB	4,B	STR$	
GOTO	B	TAB	B
IF		TAN	
INKEY$		THEN	B
INPUT	4,8,D	TO	3
INT		UNPLOT	4,B
LEN		USR	B
LET	4	VAL	C

Remarque : toutes les expressions utilisant une ou plusieurs variables non définies donnent un code erreur 2.

JEU DE CARACTERES

Tous les caractères de ce tableau sont affichables à l'écran (même les instructions), à l'exception des commandes comme EDIT ou GRAPHICS. On peut les obtenir par le programme suivant :

```
10 FOR I=0 TO 255
20 PRINT CHR$ I ; "    " ;
30 NEXT I
```

Dec.	*Hexa.*	*Caractères*	*Dec.*	*Hexa.*	*Caractères*
0	00	espace	24	18	/
1	01	▗	25	19	;
2	02	▖	26	1A	,
3	03	▄	27	1B	.
4	04	▝	28	1C	0
5	05	▐	29	1D	1
6	06	▚	30	1E	2
7	07	▟	31	1F	3
8	08	▒	32	20	4
9	09	▂	33	21	5
10	0A	▒	34	22	6
11	0B	"	35	23	7
12	0C	£	36	24	8
13	0D	$	37	25	9
14	0E	:	38	26	A
15	0F	?	39	27	B
16	10	(	40	28	C
17	11	)	41	29	D
18	12	>	42	2A	E
19	13	<	43	2B	F
20	14	=	44	2C	G
21	15	+	45	2D	H
22	16	-	46	2E	I
23	17	*	47	2F	J
			48	30	K
			49	31	L
			50	32	M
			51	33	N
			52	34	O

Dec.	*Hexa.*	*Caractères*	*Dec.*	*Hexa.*	*Caractères*
53	35	P	100	64	⎫
54	36	Q	101	65	⎪
55	37	R	102	66	⎪
56	38	S	103	67	⎪
57	39	T	104	68	⎪
58	3A	U	105	69	⎪ inutilisés
59	3B	V	106	6A	⎪
60	3C	W	107	6B	⎪
61	3D	X	108	6C	⎪
62	3E	Y	109	6D	⎪
63	3F	Z	110	6E	⎪
64	40	**RND**	111	6F	⎭
65	41	**INKEY$**	112	70	montée curseur
66	42	**PI**	113	71	descente curseur
67	43	⎫	114	72	curs.vers la gauche
68	44	⎪	115	73	curs.vers la droite
69	45	⎪	116	74	**GRAPHICS**
70	46	⎪	117	75	**EDIT**
71	47	⎪	118	76	**NEWLINE**
72	48	⎪	119	77	**RUBOUT**
73	49	⎪	120	78	mode [K] ou [L]
74	4A	⎪	121	79	**FUNCTION**
75	4B	⎪	122	7A	inutilisé
76	4C	⎪	123	7B	inutilisé
77	4D	⎪	124	7C	inutilisé
78	4E	⎪	125	7D	inutilisé
79	4F	⎪	126	7E	nombre
80	50	⎪	127	7F	curseur
81	51	⎪	128	80	■
82	52	⎪	129	81	▗
83	53	⎪ inutilisés	130	82	▖
84	54	⎪	131	83	▄
85	55	⎪	132	84	▝
86	56	⎪	133	85	▐
87	57	⎪	134	86	▚
88	58	⎪	135	87	▜
89	59	⎪	136	88	▒
90	5A	⎪	137	89	▂
91	5B	⎪	138	8A	▁
92	5C	⎪	139	8B	" inversé
93	5D	⎪			
94	5E	⎪			
95	5F	⎪			
96	60	⎪			
97	61	⎪			
98	62	⎪			
99	63	⎭			

Dec.	*Hexa.*	*Caractères*	*Dec.*	*Hexa.*	*Caractères*
140	8C	£ inversé	186	BA	U inversé
141	8D	$ inversé	187	BB	V inversé
142	8E	: inversé	188	BC	W inversé
143	8F	? inversé	189	BD	X inversé
144	90	(inversé	190	BE	Y inversé
145	91	) inversé	191	BF	Z inversé
146	92	> inversé	192	CO	" "
147	93	< inversé	193	C1	**AT**
148	94	= inversé	194	C2	**TAB**
149	95	+ inversé	195	C3	inutilisé
150	96	- inversé	196	C4	**CODE**
151	97	* inversé	197	C5	**VAL**
152	98	/ inversé	198	C6	**LEN**
153	99	; inversé	199	C7	**SIN**
154	9A	, inversé	200	C8	**COS**
155	9B	. inversé	201	C9	**TAN**
156	9C	0 inversé	202	CA	**ASN**
157	9D	1 inversé	203	CB	**ACS**
158	9E	2 inversé	204	CC	**ATN**
159	9F	3 inversé	205	CD	**LN**
160	A0	4 inversé	206	CE	**EXP**
161	A1	5 inversé	207	CF	**INT**
162	A2	6 inversé	208	D0	**SQR**
163	A3	7 inversé	209	D1	**SGN**
164	A4	8 inversé	210	D2	**ABS**
165	A5	9 inversé	211	D3	**PEEK**
166	A6	A inversé	212	D4	**USR**
167	A7	B inversé	213	D5	**STR$**
168	A8	C inversé	214	D6	**CHR$**
169	A9	D inversé	215	D7	**NOT**
170	AA	E inversé	216	D8	**
171	AB	F inversé	217	D9	**OR**
172	AC	G inversé	218	DA	**AND**
173	AD	H inversé	219	DB	< =
174	AE	I inversé	220	DC	> =
175	AF	J inversé	221	DD	< >
176	B0	K inversé	222	DE	**THEN**
177	B1	L inversé	223	DF	**TO**
178	B2	M inversé	224	E0	**STEP**
179	B3	N inversé	225	E1	**LPRINT**
180	B4	O inversé	226	E2	**LLIST**
181	B5	P inversé	227	E3	**STOP**
182	B6	Q inversé	228	E4	**SLOW**
183	B7	R inversé	229	E5	**FAST**
184	B8	S inversé	230	E6	**NEW**
185	B9	T inversé	231	E7	SCROLL

Dec.	*Hexa.*	*Caractères*
232	E8	**CONT**
233	E9	**DIM**
234	EA	**REM**
235	EB	**FOR**
236	EC	**GOTO**
237	ED	**GOSUB**
238	EE	**INPUT**
239	EF	**LOAD**
240	F0	**LIST**
241	F1	**LET**
242	F2	**PAUSE**
243	F3	**NEXT**
244	F4	POKE
245	F5	**PRINT**
246	F6	PLOT
247	F7	**RUN**
248	F8	SAVE
249	F9	**RAND**
250	FA	IF
251	FB	**CLS**
252	FC	UNPLOT
253	FD	**CLEAR**
254	FE	RETURN
255	FF	**COPY**

Le code de la dernière touche appuyée est stocké dans la mémoire système 4025 (16422 dec.). La routine du Basic qui scrute le clavier est en 02BB (699 dec.) et celle qui décode en 07BD (1981 dec.).

Codes dans l'ordre du clavier

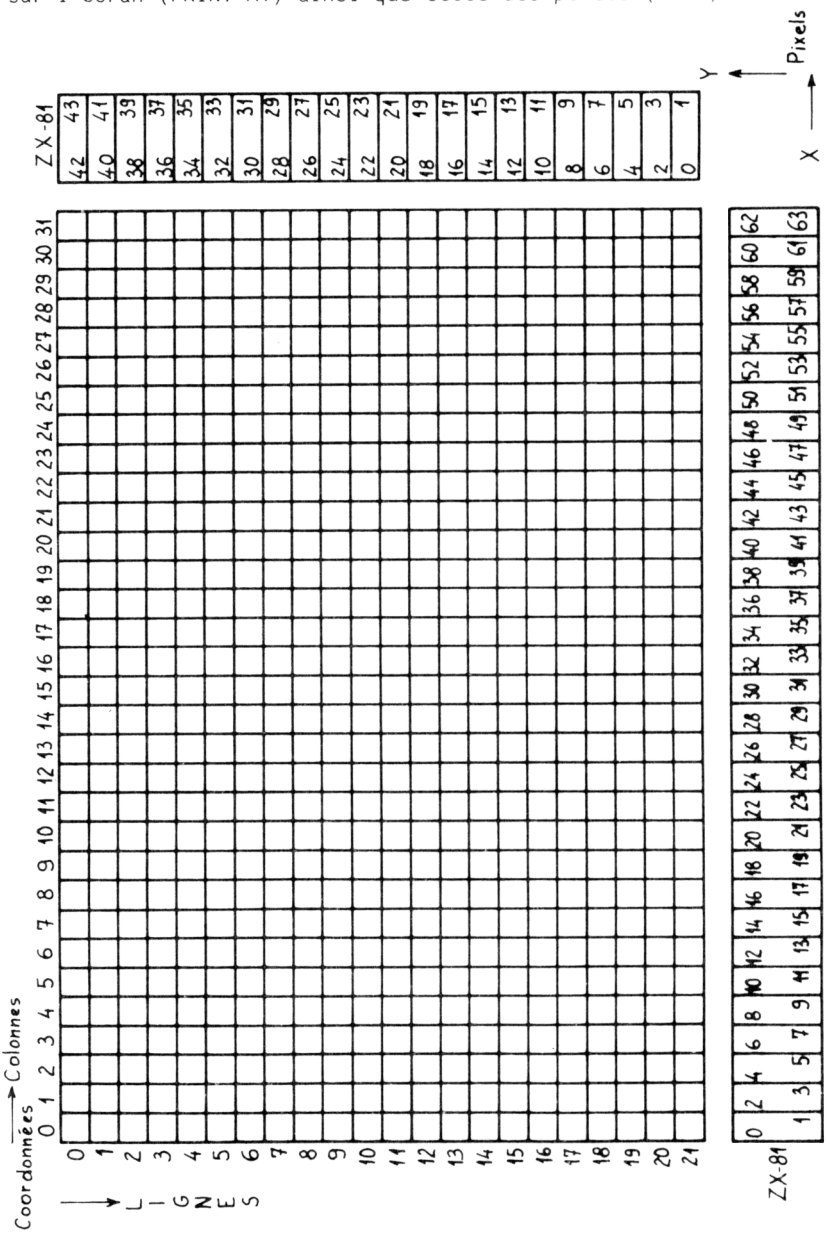
Touche sans SHIFT			*Touche avec SHIFT*		
Touche	*Dec.*	*Hexa.*	*Touche*	*Dec.*	*Hexa.*
1	65015	FDF7	EDIT	64759	FCF7
2	64503	FBF7	AND	64247	FAF7
3	63479	F7F7	THEN	63223	F6F7
4	61431	EFF7	TO	61175	EEF7
5	57335	DFF7	<--	57079	DEF7
6	57327	DFEF	V	57071	DEEF
7	61423	EFEF	^	61167	EEEF
8	63471	F7EF	-->	63215	F6EF
9	64495	FBEF	GRAPHICS	64239	FAEF
0	65007	FDEF	RUBOUT	64751	FCEF
Q	65019	FDFB	""	64763	FCFB
W	64507	FBFB	OR	64251	FAFB
E	63438	F7FB	STEP	63227	F6FB
R	61435	EFFB	<=	61179	EEFB
T	57339	DFFB	<>	57083	DEFB
Y	57311	DFDF	>=	57055	DEDF
U	61047	EFDF	$	61151	EEDF
I	63455	F7DF	(	63199	F6DF
O	64479	FBDF	)	64223	FADF
P	64991	FDDF	"	64735	FCDF
A	65021	FDFD	STOP	64765	FCFD
S	64509	FBFD	LPRINT	64253	FAFD
D	63485	F7FD	SLOW	63229	F6FD
F	61437	EFFD	FAST	61181	EEFD
G	57341	DFFD	LLIST	57085	DEFD
H	57279	DFBF	**	57023	DEBF
J	61375	EFBF	-	61119	EEBF
K	63423	F7BF	+	63167	F6BF
+	64447	FBBF	=	64191	FABF
NEW LINE	64959	FDBF	FUNCTION	64959	FCBF
Z	64510	FBFE	:	64254	FAFE
X	63486	F7FE	;	63230	F6FE
C	61438	EFFE	?	61182	EEFE
V	57342	DFFE	/	57086	DEFE
B	57215	DF7F	*	56959	DE7F

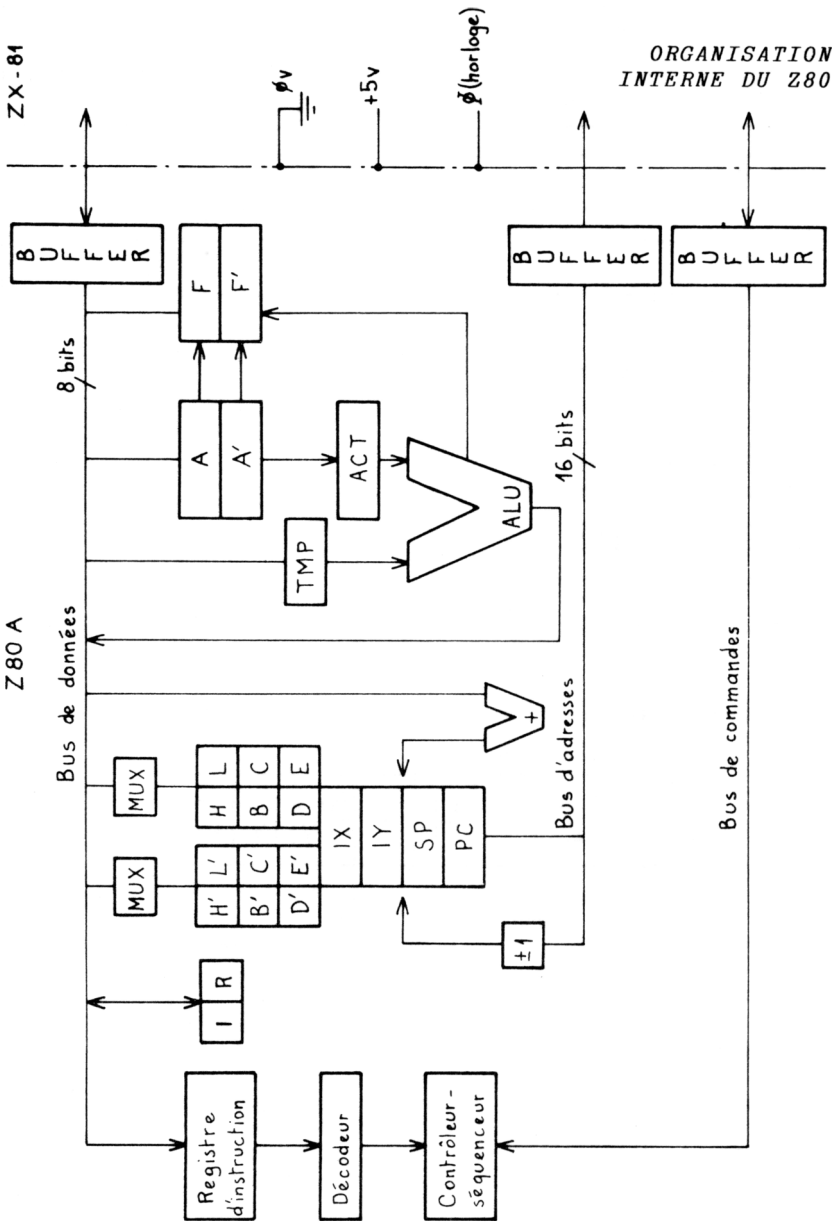
CODES CLAVIER

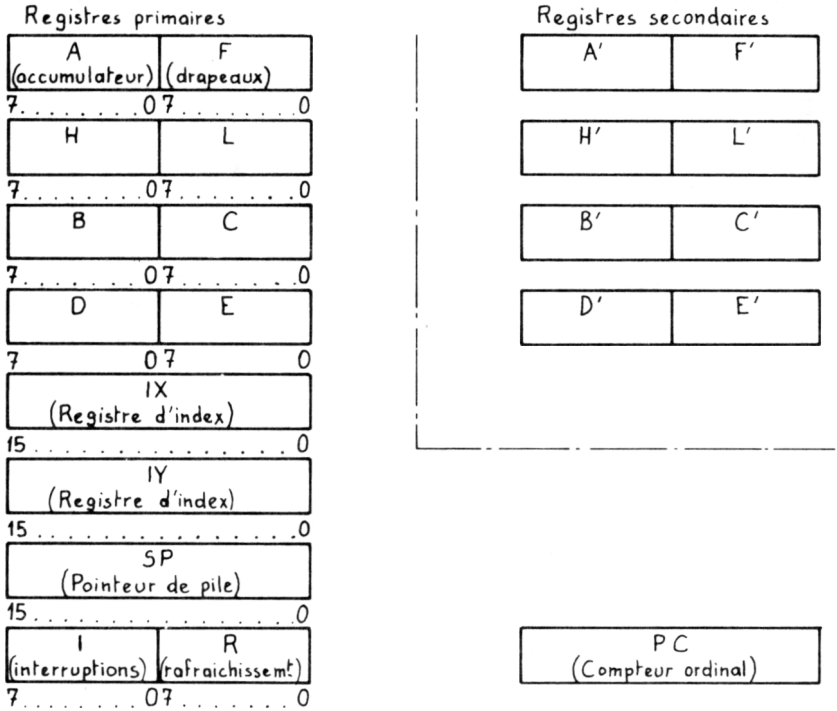
Touche sans SHIFT			*Touche avec SHIFT*		
Touche	*Dec.*	*Hexa.*	*Touche*	*Dec.*	*Hexa.*
N	61311	EF7F	<	61055	EE7F
M	63359	E77F	>	63103	F67F
.	64383	FB7F	,	64127	FA7F
SPACE	64895	FD7F	£	64639	FC7F

Remarque : aucune touche appuyée : 65535 (FFFF en hexa.).

La grille ci-dessous vous donne la position des caractères sur l'écran (PRINT AT) ainsi que celle des pixels (PLOT).




ORGANISATION INTERNE DU Z80
ZX-81
Øv
+5v
Ø (horloge)
BUFFER
BUFFER
BUFFER
8 bits
16 bits
F
F'
A
A'
ACT
TMP
ALU
Z80 A
Bus de données
Bus d'adresses
Bus de commandes
MUX
MUX
H L
B C
D E
H' L'
B' C'
D' E'
IX
IY
SP
PC
±1
I R
Registre d'instruction
Décodeur
Contrôleur - séquenceur



Détails du registre F (flag = drapeau)

7	6	5	4	3	2	1	0
S	Z	-	H	-	P/V	N	C

S = Signe : passe à 1 si le bit le plus significatif du résultat d'une opération est 1.

Z = Zéro : passe à 1 si le résultat de l'opération est nul.

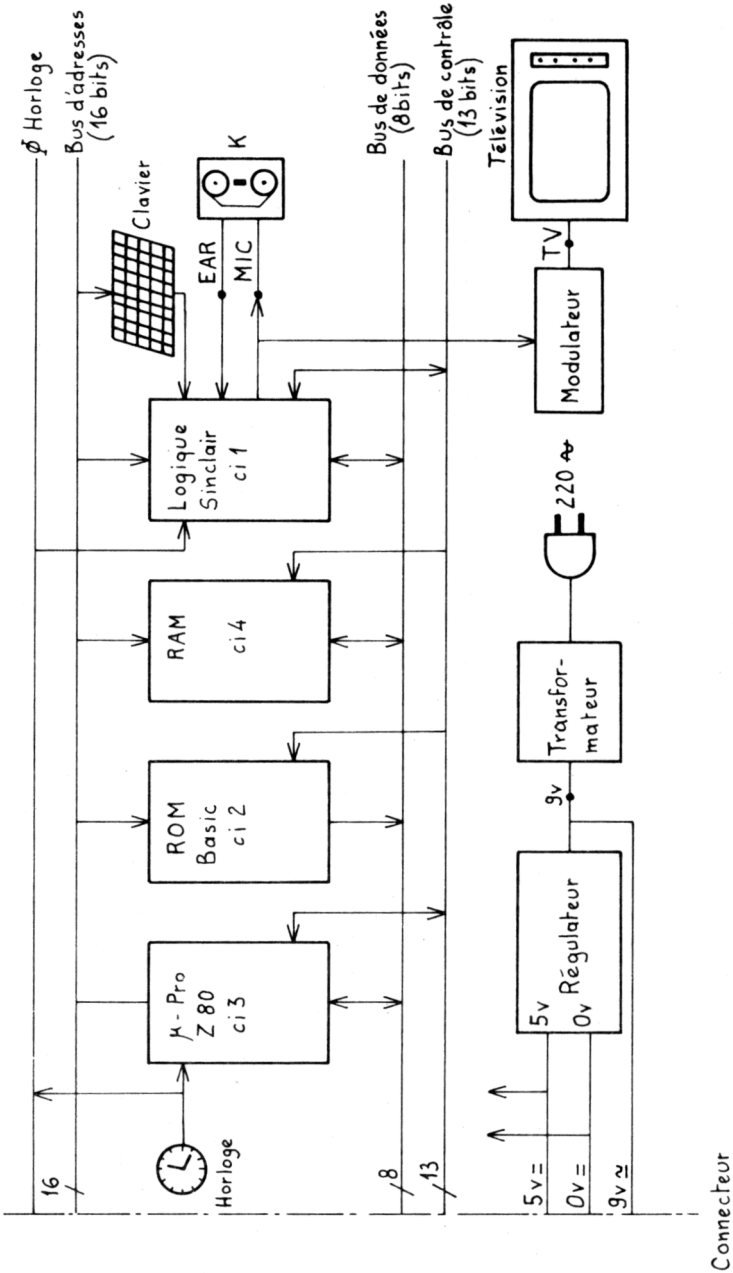
H = Demi : identique à C, mais pour les opérations sur des demi-octets.

P/V = Parité/dépassement : P=1 s'il y a un nombre pair de bits à 1, ou V=1 s'il y a dépassement de capacité après une opération avec des nombres signés.

N = Opération précédente : N=1 si l'opération précédente était une soustraction, et N=0 pour une addition.

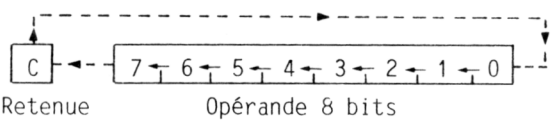
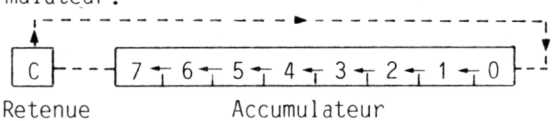
C = Retenue : passe à 1 si le résultat nécessite une retenue (soustraction) ou un report (addition).

Remarque : les drapeaux H et N ne peuvent pas être testés.

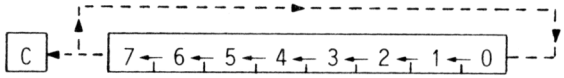

ϕ Horloge
Bus d'adresses (16 bits)
Clavier
K
EAR
MIC
Bus de données (8bits)
Bus de contrôle (13 bits)
Télévision
TV
Modulateur
Logique Sinclair ci 1
RAM ci 4
ROM Basic ci 2
μ-Pro Z 80 ci 3
220 v
Transfor-mateur
9v
5v
0v Régulateur
Horloge
16
8
13
5v =
0v =
9v ≃
Connecteur arrière

Mnémonique	*Opération effectuée*
ADC	Addition avec retenue.
ADD	Addition sans retenue.
BIT	Teste un bit particulier d'un octet.
CALL cc,nn	Appel conditionnel d'un sous-programme.
CALL	Appel inconditionnel d'un sous-programme.
CCF	Complémente l'indicateur de retenue.
CP	Compare l'opérande et l'acucmulateur.
CPD	Compare le contenu de l'adresse pointée par HL. Décrémente HL et BC.
CPDR	Compare le contenu de l'adresse pointée par HL. Décrémente HL et BC. Répète la séquence jusqu'à ce que BC=0.
CPI	Compare le contenu de l'adresse pointée par HL. Incrémente HL et décrémente BC.
CPIR	Compare le contenu de l'adresse pointée par HL. Incrémente HL et décrémente BC. Répète la séquence jusqu'à ce que BC=0.
CPL	Complémente l'accumulateur.
DAA	Ajustement décimal de l'accumulateur.
DEC	Décrémente un registre, une paire de registres ou une adresse pointée par HL.
DI	Désactive les interruptions.
DJNZ	Décrémente B et effectue un saut relatif si B≠0.
EI	Active les interruptions.
EX	Echange les contenus des registres.
EXX	Echange le contenu des registres BC, DE et HL avec les registres BC', DE' et HL'.
HALT	Place le microprocesseur en position d'attente d'une interruption ou d'un reset.
IM	Positionne un des trois modes d'interruption (de 0 à 2).
IN	Charge l'accumulateur ou un registre avec le contenu d'un port d'entrées/sorties.
INC	Incrémente un registre, une paire de registres ou le contenu de l'adresse pointée par HL.

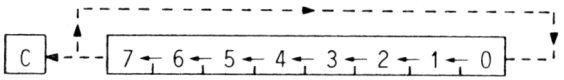
Mnémonique	*Opération effectuée*
IND	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et décrémente HL et B.
INDR	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et décrémente HL et B. Répète la séquence jusqu'à ce que B=0.
INI	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et incrémente HL et décrémente B.
INIR	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et incrémente HL et décrémente B. Répète la séquence jusqu'à ce que B=0.
JP	Saut inconditionnel à l'adresse donnée ou à celle pointée par HL, IX et IY.
JP cc,aa	Saut conditionnel (cc) à l'adresse donnée (aa).
JR e	Saut inconditionnel relatif à PC plus déplacement (e).
JR cc,e	Saut conditionnel (cc) relatif à PC plus déplacement (e).
LD	Charge l'accumulateur, un registre ou une adresse avec le contenu de l'accumulateur, un registre ou une adresse.
LDD	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis décrémente DE, HL et BC.
LDDR	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis décrémente DE, HL et BC. Répète la séquence jusqu'à ce que BC=0.
LDI	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis incrémente DE et HL et décrémente BC.
LDIR	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis incrémente DE et HL et décrémente BC. Répète la séquence jusqu'à ce que BC=0.
NEG	Inverse le signe de l'accumulateur.
NOP	Le Z80 n'effectue pas d'instruction.

Mnémonique	*Opération effectuée*
OR	"Ou" logique entre l'opérande et l'accumulateur.
OTDR	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis décrémente HL et B. Répète la séquence jusqu'à ce que B=0.
OTIR	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis incrémente HL et décrémente B. Répète la séquence jusqu'à ce que B=0.
OUT	Charge le port d'entrées/sorties spécifié avec le contenu de l'accumulateur.
OUTD	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis décrémente HL et B.
OUTI	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis incrémente HL et décrémente B.
POP	Charge une paire de registres ou un index avec la dernière valeur de la pile (pointée par SP).
PUSH	Place le contenu d'une paire de registres ou d'un index dans la pile (pointé par SP).
RES	Place 0 dans le bit spécifié de l'opérande.
RET	Retour d'un sous-programme.
RETI	Retour d'un sous-programme d'interruption.
RETN	Retour d'un sous-programme d'interruption non masquable.
RL	Rotation à gauche à travers la retenue de l'opérande. 
RLA	Rotation à gauche à travers la retenue de l'accumulateur. 

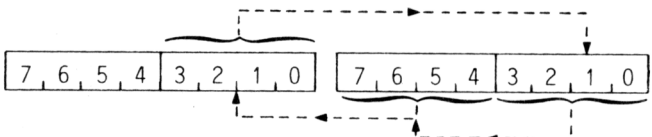
Mnémonique	*Opération effectuée*
RLC	Rotation circulaire à gauche du contenu d'un registre ou d'une adresse pointée par HL ou par IX et IY plus déplacement.
RLCA	Rotation circulaire à gauche de l'accumulateur.
RLD	Rotation BCD (binaire codé décimal) de 4 bits à gauche entre l'accumulateur et l'adresse pointée par HL.
RR	Rotation à droite à travers la retenue de l'opérande.
RRA	Rotation à droite à travers la retenue de l'accumulateur.



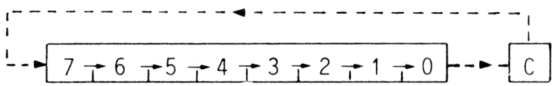
Retenue Opérande 8 bits



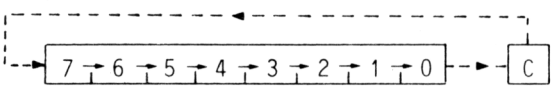
Retenue Accumulateur



Accumulateur Adresse pointée par HL

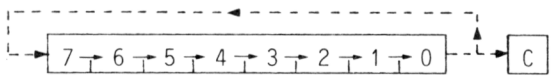


Opérande 8 bits Retenue

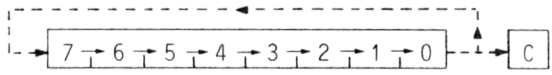


Accumulateur Retenue

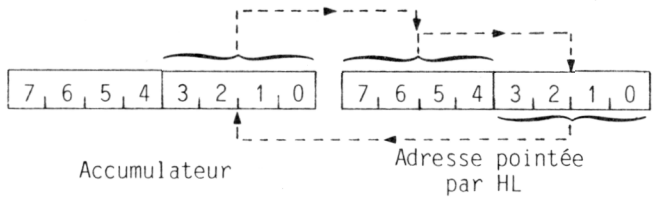
Mnémonique	*Opération effectuée*
RRC	Rotation circulaire à droite de l'opérande.



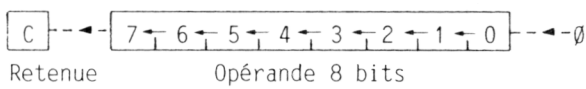
RRCA	Rotation circulaire à droite de l'accumulateur.



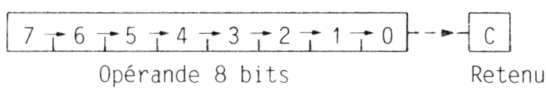
RRD	Rotation BCD à droite de 4 bits entre l'accumulateur et le contenu de l'adresse pointée par HL.



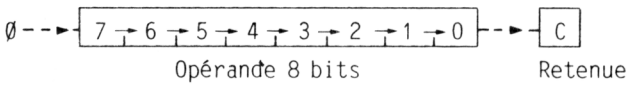
RST	Saut à une adresse sur un octet.
SBC	Soustraction avec retenue entre l'accumulateur et l'opérande ou HL et une paire de registres.
SCF	Met à 1 le drapeau de retenue.
SET	Met à 1 un bit particulier d'un registre ou d'une adresse pointée par HL ou IX et IY plus déplacement.
SLA	Décalage arithmétique à gauche de l'opérande.



SRA	Décalage arithmétique à droite de l'opérande.



NB : le bit 7 reste inchangé.

Mnémonique	*Opération effectuée*
SRL	Décalage logique à droite de l'opérande. 
SUB	Soustraction à l'accumulateur de l'opérande.
XOR	"Ou" exclusif entre l'opérande et l'accumulateur.

- d = donnée sur 8 bits
- dd = donnée sur 16 bits
- aa = adresse sur 16 bits
- ● = drapeau modifié
- 0 = drapeau à 0
- 1 = drapeau à 1

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
8E	ADC A,(HL)	●	●	●	●
DD8Ed	ADC A,(IX+d)	●	●	●	●
FD8Ed	ADC A,(IY+d)	●	●	●	●
8F	ADC A,A	●	●	●	●
88	ADC A,B	●	●	●	●
89	ADC A,C	●	●	●	●
8A	ADC A,D	●	●	●	●
8B	ADC A,E	●	●	●	●
8C	ADC A,H	●	●	●	●
8D	ADC A,L	●	●	●	●
CEd	ADC A,d	●	●	●	●
ED4A	ADC HL,BC	●	●	●	●
ED5A	ADC HL,DE	●	●	●	●
ED6A	ADC HL,HL	●	●	●	●
ED7A	ADC HL,SP	●	●	●	●
86	ADD A,(HL)	●	●	●	●
DD86d	ADD A,(IX+d)	●	●	●	●
FD86d	ADD A,(IY+d)	●	●	●	●
87	ADD A,A	●	●	●	●
80	ADD A,B	●	●	●	●
81	ADD A,C	●	●	●	●
82	ADD A,D	●	●	●	●
83	ADD A,E	●	●	●	●
84	ADD A,H	●	●	●	●
85	ADD A,L	●	●	●	●
C6d	ADD A,d	●	●	●	●
09	ADD HL,BC				●
19	ADD HL,DE				●
29	ADD HL,HL				●
39	ADD HL,SP				●
DD09	ADD IX,BC				●
DD19	ADD IX,DE				●
DD29	ADD IX,IX				●
DD39	ADD IX,SP				●
D09	ADD IY,BC				●
FD19	ADD IY,DE				●
FD29	ADD IY,IY				●
FD39	ADD IY,SP				●
A6	AND (HL)	●	●	●	0
DDA6d	AND (IX+d)	●	●	●	0
FDA6d	AND (IY+d)	●	●	●	0
A7	AND A	●	●	●	0

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
A0	AND B	●	●	●	0
A1	AND C	●	●	●	0
A2	AND D	●	●	●	0
A3	AND E	●	●	●	0
A4	AND H	●	●	●	0
A5	AND L	●	●	●	0
E6d	AND d	●	●	●	0
CB46	BIT 0,(HL)	●	●	●	
DDCBd46	BIT 0,(IX+d)	●	●	●	
FDCBd46	BIT 0,(IY+d)	●	●	●	
CB47	BIT 0,A	●	●	●	
CB40	BIT 0,B	●	●	●	
CB41	BIT 0,C	●	●	●	
CB42	BIT 0,D	●	●	●	
CB43	BIT 0,E	●	●	●	
CB44	BIT 0,H	●	●	●	
CB45	BIT 0,L	●	●	●	
CB4E	BIT 1 (HL)	●	●	●	
DDCBd4E	BIT 1,(IX+d)	●	●	●	
FDCBd4E	BIT 1,(IY+d)	●	●	●	
CB4F	BIT 1,A	●	●	●	
CB48	BIT 1,B	●	●	●	
CB49	BIT 1,C	●	●	●	
CB4A	BIT 1,D	●	●	●	
CB4B	BIT 1,E	●	●	●	
CB4C	BIT 1,H	●	●	●	
CB4D	BIT 1,L	●	●	●	
CB56	BIT 2,(HL)	●	●	●	
DDCBd56	BIT 2,(IX+d)	●	●	●	
FDCBd56	BIT 2,(IY+d)	●	●	●	
CB57	BIT 2,A	●	●	●	
CB50	BIT 2,B	●	●	●	
CB51	BIT 2,C	●	●	●	
CB52	BIT 2,D	●	●	●	
CB53	BIT 2,E	●	●	●	
CB54	BIT 2,H	●	●	●	
CB55	BIT 2,L	●	●	●	
CB5E	BIT 3,(HL)	●	●	●	
DDCBd5E	BIT 3,(IX+d)	●	●	●	
FDCBd5E	BIT 3,(IY+d)	●	●	●	
CB5F	BIT 3,A	●	●	●	
CB58	BIT 3,B	●	●	●	
CB59	BIT 3,C	●	●	●	
CB5A	BIT 3,D	●	●	●	
CB5B	BIT 3,E	●	●	●	
CB5C	BIT 3,H	●	●	●	
CB5D	BIT 3,L	●	●	●	

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
CB66	BIT 4,(HL)	●	●	●	
DDCBd66	BIT 4,(IX+d)	●	●	●	
FDCBd66	BIT 4,(IY+d)	●	●	●	
CB67	BIT 4,A	●	●	●	
CB60	BIT 4,B	●	●	●	
CB61	BIT 4,C	●	●	●	
CB62	BIT 4,D	●	●	●	
CB63	BIT 4,E	●	●	●	
CB64	BIT 4,H	●	●	●	
CB65	BIT 4,L	●	●	●	
CB6E	BIT 5,(HL)	●	●	●	
DDCBd6E	BIT 5,(IX+d)	●	●	●	
FDCBd6E	BIT 5,(IY+d)	●	●	●	
CB6F	BIT 5,A	●	●	●	
CB68	BIT 5,B	●	●	●	
CB69	BIT 5,C	●	●	●	
C86A	BIT 5,D	●	●	●	
CB6B	BIT 5,E	●	●	●	
CB6C	BIT 5,H	●	●	●	
CB6D	BIT 5,L	●	●	●	
DB76	BIT 6,(HL)	●	●	●	
DDCBd76	BIT 6,(IX+d)	●	●	●	
FDCBd76	BIT 6,(IY+d)	●	●	●	
CB77	BIT 6,A	●	●	●	
CB70	BIT 6,B	●	●	●	
CB71	BIT 6,C	●	●	●	
CB72	BIT 6,D	●	●	●	
CB73	BIT 6,E	●	●	●	
CB74	BIT 6,H	●	●	●	
CB75	BIT 6,L	●	●	●	
CB7E	BIT 7,(HL)	●	●	●	
DDCBd7E	BIT 7,(IX+d)	●	●	●	
FDCBd7E	BIT 7,(IY+d)	●	●	●	
CB7F	BIT 7,A	●	●	●	
CB78	BIT 7,B	●	●	●	
CB79	BIT 7,C	●	●	●	
CB7A	BIT 7,D	●	●	●	
CB7B	BIT 7,E	●	●	●	
CB7C	BIT 7,H	●	●	●	
CB7D	BIT 7,L	●	●	●	
DCaa	CALL C,aa				
FCaa	CALL M,aa				
D4aa	CALL NC,aa				
C4aa	CALL NZ,aa				
F4dd	CALL P,dd				
ECdd	CALL PE,dd				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
E4dd	CALL PO,dd				
CCdd	CALL Z,dd				
CDdd	CALL dd				
3F	CCF				●
BE	CP (HL)	●	●	●	●
DDBEd	CP (IX+d)	●	●	●	●
FDBEd	CP (IY+d)	●	●	●	●
BF	CP A	●	●	●	●
B8	CP B	●	●	●	●
B9	CP C	●	●	●	●
BA	CP D	●	●	●	●
BB	CP E	●	●	●	●
BC	CP H	●	●	●	●
BD	CP L	●	●	●	●
FEd	CP d	●	●	●	●
EDA9	CPD	●	●	●	
EDB9	CPDR	●	●	●	
ED81	CPIR	●	●	●	
EDA1	CPI	●	●	●	
2F	CPL				
27	DAA	●	●	●	
35	DEC (HL)	●	●	●	
DD35d	DEC (IX+d)	●	●	●	
FD35d	DEC (IY+d)	●	●	●	
3D	DEC A	●	●	●	
05	DEC B	●	●	●	
0B	DEC BC				
0D	DEC C	●	●	●	
15	DEC D	●	●	●	
1B	DEC DE				
1D	DEC E	●	●	●	
25	DEC H	●	●	●	
28	DEC HL				
DD2B	DEC IX				
FD2B	DEC IY				
2D	DEC L	●	●	●	
3B	DEC SP				
F3	DI				
10d	DJNZ d				
FB	EI				
E3	EX (SP),HL				
DDE3	EX (SP),IX				
FDE3	EX (SP),IY				
08	EX AF,AF'				
EB	EX DE,HL				
D9	EXX				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
76	HALT				
ED46	IM 0				
ED56	IM 1				
ED5E	IM 2				
ED78	IN A,(C)	●	●	●	
ED40	IN B,(C)	●	●	●	
ED48	IN C,(C)	●	●	●	
ED50	IN D,(C)	●	●	●	
ED58	IN E,(C)	●	●	●	
ED60	IN H,(C)	●	●	●	
ED68	IN L,(C)	●	●	●	
34	INC (HL)	●	●	●	
DD34d	INC (IX+d)	●	●	●	
FD34d	INC (IY+d)	●	●	●	
3C	INC A	●	●	●	
04	INC B	●	●	●	
03	INC BC				
0C	INC C	●	●	●	
14	INC D	●	●	●	
13	INC DE				
1C	INC E	●	●	●	
24	INC H	●	●	●	
23	INC HL				
DD23	INC IX				
FD23	INC IY				
2C	INC L	●	●	●	
33	INC SP				
DBd	IN A,(d)				
EDAA	IND	●	●	●	
EDBA	INDR	●	●	●	
EDA2	INI	●	●	●	
EDB2	INIR	●	●	●	
C3dd	JP dd				
E9	JP (HL)				
DDE9	JP (IX)				
FDE9	JP (IY)				
DAdd	JP C,aa				
FAaa	JP M,aa				
D2aa	JP NC,aa				
C2aa	JP NZ,aa				
F2aa	JP P,aa				
EAaa	JP PE,aa				
E2aa	JP PO,aa				
CAaa	JP Z,aa				
38d	JR C,d				
30d	JR NC,d				
20d	JR NZ,d				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
28d	JR Z,d				
18d	JR d				
02	LD (BC),A				
12	LD (DE),A				
77	LD (HL),A				
70	LD (HL),B				
71	LD (HL),C				
72	LD (HL),D				
73	LD (HL),E				
74	LD (HL),H				
75	LD (HL),L				
36d	LD (HL),d				
DD77d	LD (IX+d),A				
DD70d	LD (IX+d),B				
DD71d	LD (IX+d),C				
DD72d	LD (IX+d),D				
DD73d	LD (IX+d),E				
DD74d	LD (IX+d),H				
DD75d	LD (IX+d),L				
DD36d20	LD (IX+d),d				
FD77d5	LD (IY+d),A				
FD70d	LD (IY+d),B				
FD71d	LD (IY+d),C				
FD72d	LD (IY+d),D				
FD73d	LD (IY+d),E				
FD74d	LD (IY+d),H				
FD75d	LD (IY+d),L				
FD36d20	LD (IY+d),d				
32dd	LD (dd),A				
ED43dd	LD (dd),BC				
ED53dd	LD (dd),DE				
22dd	LD (dd),HL				
DD22dd	LD (dd),IX				
FD22dd	LD (dd),IY				
ED73dd	LD (dd),SP				
0A	LD A,(BC)				
1A	LD A,(DE)				
7E	LD A,(HL)				
DD7Ed	LD A,(IX+d)				
FD7Ed	LD A,(IY+d)				
3Add	LD A,(dd)				
7F	LD A,A				
78	LD A,B				
79	LD A,C				
7A	LD A,D				
7B	LD A,E				
7C	LD A,H				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
ED57	LD A,I	●	●	●	
7D	LD A,L				
3E	LD A,d				
ED5F	LD A,R	●	●	●	
46	LD B,(HL)				
DD46d	LD B,(IX+d)				
FD46d	LD B,(IY+d)				
47	LD B,A				
40	LD B,B				
41	LD B,C				
42	LD B,D				
43	LD B,E				
44	LD B,H				
45	LD B,L				
06d	LD B,d				
ED4Bdd	LD BC,(dd)				
01dd	LD NC,dd				
4E	LD C,(HL)				
DD4Ed	LD C,(IX+d)				
FD4Ed	LD C,(IY+d)				
4F	LD C,A				
48	LD C,B				
49	LD C,C				
4A	LD C,D				
4B	LD C,E				
4C	LD C,H				
4D	LD C,L				
0Ed	LD C,d				
56	LD D,(HL)				
DD56d	LD D,(IX+d)				
FD56d	LD D,(IY+d)				
57	LD D,A				
50	LD D,B				
51	LD D,C				
52	LD D,D				
53	LD D,E				
54	LD D,H				
55	LD D,L				
16d	LD D,d				
ED5Bdd	LD DE,(dd)				
11dd	LD DE,dd				
5E	LD E,(HL)				
DD5Ed	LD E,(IX+d)				
FD5Ed	LD E,(IY+d)				
5F	LD E,A				
58	LD E,B				
59	LD E,C				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
5A	LD E,D				
5B	LD E,E				
5C	LD E,H				
5D	LD E,L				
1E20	LD E,n				
66	LD H,(HL)				
DD66d	LD H,(IX+d)				
FD66d	LD H,(IY+d)				
67	LD H,A				
60	LD H,B				
61	LD H,C				
62	LD H,D				
63	LD H,E				
64	LD H,H				
65	LD H,L				
26d	LD H,d				
2Add	LD HL,(dd)				
21dd	LD HL,dd				
ED47	LD I,A				
DD2Add	LD IX,(dd)				
DD21dd	LD IX,dd				
FD2Add	LD IY,(dd)				
FD21dd	LD IY,dd				
6E	LD L,(HL)				
DD6Ed	LD L,(IX+d)				
FD6Ed	LD L,(IY+d)				
6F	LD L,A				
68	LD L,B				
69	LD L,C				
6A	LD L,D				
6B	LD L,E				
6C	LD L,H				
6D	LD L,L				
2Ed	LD L,d				
ED4F	LD R,A				
ED7Bdd	LD SP,(dd)				
F9	LD SP,HL				
DDF9	LD SP,IX				
FDF9	LD SP,IY				
31dd	LD SP,dd				
EDA8	LDD			●	
EDB8	LDDR			0	
EDA0	LDI			●	
EDB0	LDIR			0	
ED44	NEG				
00	NOP				
B6	OR (HL)	●	●	●	0

CODES DES INSTRUCTIONS Z80
PAR ORDRE ALPHABETIQUE

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
DDB6d	OR (IX+d)	●	●	●	0
FDB6d	OR (IY+d)	●	●	●	0
B7	OR A	●	●	●	0
B0	OR B	●	●	●	0
B1	OR C	●	●	●	0
B2	OR D	●	●	●	0
B3	OR E	●	●	●	0
B4	OR H	●	●	●	0
B5	OR L	●	●	●	0
F6d	OR d	●	●	●	0
ED8B	OTDR	●	●	●	
EDB3	OTIR	●	●	●	
ED79	OUT (C),A				
ED41	OUT (C),B				
ED49	OUT (C),C				
ED51	OUT (C),D				
ED59	OUT (C),E				
ED61	OUT (C),H				
ED69	OUT (C),L				
D3d	OUT (d),A				
EDAB	OUTD	●	●	●	
EDA3	OUTI	●	●	●	
F1	POP AF				
C1	POP BC				
D1	POP DE				
E1	POP HL				
DDE1	POP IX				
FDE1	POP IY				
F5	PUSH AF				
C5	PUSH BC				
D5	PUSH DE				
E5	PUSH HL				
DDE5	PUSH IX				
FDE5	PUSH IY				
CB86	RES 0,(HL)				
DDCBd86	RES 0,(IX+d)				
FDCBd86	RES 0,(IY+d)				
DB87	RES 0,A				
DB80	RES 0,B				
CB81	RES 0,C				
CB82	RES 0,D				
CB83	RES 0,E				
CB84	RES 0,H				
CB85	RES 0,L				
CB8E	RES 1,(HL)				
DDCBd8E	RES 1,(IX+d)				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
FDCBd8E	RES 1,(IY+d)				
CB8F	RES 1,A				
CB88	RES 1,B				
CB89	RES 1,C				
CB8A	RES 1,D				
CB8B	RES 1,E				
CB8C	RES 1,H				
CB8D	RES 1,L				
CB96	RES 2,(HL)				
DDCBd96	RES 2,(IX+d)				
FDCBd96	RES 2,(IY+d)				
CB97	RES 2,A				
CB90	RES 2,B				
CB91	RES 2,C				
CB92	RES 2,D				
CB93	RES 2,E				
CB94	RES 2,H				
CB95	RES 2,L				
CB9E	RES 3,(HL)				
DDCBd9E	RES 3,(IX+d)				
FDCBd9E	RES 3,(IY+d)				
CB9F	RES 3,A				
CB98	RES 3;B				
CB99	RES 3,C				
CB9A	RES 3,D				
CB9B	RES 3,E				
CB9C	RES 3,H				
CB9D	RES 3,L				
CBA6	RES 4,(HL)				
DDCBdA6	RES 4,(IX+d)				
FDCBdA7	RES 4,(IY+d)				
CBA7	RES 4,A				
CBA0	RES 4,B				
CBA1	RES 4,C				
CBA2	RES 4,D				
DBA3	RES 4,E				
CBA4	RES 4,H				
CBA5	RES 4,L				
CBAE	RES 5,(HL)				
DDCBdAE	RES 5,(IX+d)				
FDCBdAE	RES 5,(IY+d)				
CBAF	RES 5,A				
CBA8	RES 5,B				
CBA9	RES 5,C				
CBAA	RES 5,D				
CBAB	RES 5,E				
CBAC	RES 5,H				

CODES DES INSTRUCTIONS Z80
PAR ORDRE ALPHABETIQUE

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
CBAD	RES 5,L				
CBB6	RES 6,(HL)				
DDCBdB6	RES 6,(IX+d)				
FDCBdB6	RES 6,(IY+d)				
CBB7	RES 6,A				
CBB0	RES 6,B				
CBB1	RES 6,C				
CBB2	RES 6,D				
CBB3	RES 6,E				
CBB4	RES 6,H				
CBB5	RES 6,L				
CBBE	RES 7,(HL)				
DDCBdBE	RES 7,(IX+d)				
FDCBdBE	RES 7,(IY+d)				
CBBF	RES 7,A				
CBB8	RES 7,B				
CBB9	RES 7,C				
CBBA	RES 7,D				
CBBB	RES 7,E				
CBBC	RES 7,H				
CBBD	RES 7,L				
C9	RET				
D8	RET C				
F8	RET M				
D0	RET NC				
C0	RET NZ				
F0	RET P				
E8	RET PE				
E0	RET PO				
C8	RET Z				
ED4D	RETI				
ED45	RETN				
CB16	RL (HL)	●	●	●	●
DDCBd16	RL (IX+d)	●	●	●	●
FDCBd16	RL (IY+d)	●	●	●	●
CB17	RL A	●	●	●	●
CB10	RL B	●	●	●	●
CB11	RL C	●	●	●	●
CB12	RL D	●	●	●	●
CB13	RL E	●	●	●	●
CB14	RL H	●	●	●	●
CB15	RL L	●	●	●	●
17	RLA				●
CB06	RLC (HL)	●	●	●	●
DDCBd06	RLC (IX+d)	●	●	●	●
FDCBd06	RLC (IY+d)	●	●	●	●

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
CB07	RLC A	●	●	●	●
CB00	RLC B	●	●	●	●
CB01	RLC C	●	●	●	●
CB02	RLC D	●	●	●	●
CB03	RLC E	●	●	●	●
CB04	RLC H	●	●	●	●
CB05	RLC L	●	●	●	●
07	RLCA				●
ED6F	RLD	●	●	●	
CB1E	RR (HL)	●	●	●	●
DDCBd1E	RR (IX+d)	●	●	●	●
FDCBd1E	RR (IY+d)	●	●	●	●
CB1F	RR A	●	●	●	●
CB18	RR B	●	●	●	●
CB19	RR C	●	●	●	●
CB1A	RR D	●	●	●	●
CB1B	RR E	●	●	●	●
CB1C	RR H	●	●	●	●
CB1D	RR L	●	●	●	●
1F	RRA				●
CB0E	RRC (HL)	●	●	●	●
DDCBd0E	RRC (IX+d)	●	●	●	●
FDCBd0E	RRC (IY+d)	●	●	●	●
CB0F	RRC A	●	●	●	●
CB08	RRC B	●	●	●	●
CB09	RRC C	●	●	●	●
CB0A	RRC D	●	●	●	●
CB0B	RRC E	●	●	●	●
CB0C	RRC H	●	●	●	●
CB0D	RRC L	●	●	●	●
0F	RRCA				●
ED67	RRD	●	●	●	
C7	RST 00H				
CF	RST 08H				
D7	RST 10H				
DF	RST 18H				
E7	RST 20H				
EF	RST 28H				
F7	RST 30H				
FF	RST 38H				
DEd	SBC A,d	●	●	●	●
9E	SBC A,(HL)	●	●	●	●
DD9Ed	SBC A,(IX+d)	●	●	●	●
FD9Ed	SBC A,(IY+d)	●	●	●	●
9F	SBC A,A	●	●	●	●
98	SBC A,B	●	●	●	●

CODES DES INSTRUCTIONS Z80
PAR ORDRE ALPHABETIQUE

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
99	SBC A,C	●	●	●	●
9A	SBC A,D	●	●	●	●
9B	SBC A,E	●	●	●	●
9C	SBC A,H	●	●	●	●
9D	SBC A,L	●	●	●	●
ED42	SBC HL,BC	●	●	●	●
ED52	SBC HL,DE	●	●	●	●
ED62	SBC HL,HL	●	●	●	●
ED72	SBC HL,SP	●	●	●	●
37	SCF				1
CBC6	SET 0,(HL)				
DDCBdC6	SET 0,(IX+d)				
FDCBdC6	SET 0,(IY+d)				
CBC7	SET 0,A				
CBC0	SET 0,B				
CBC1	SET 0,C				
CBC2	SET 0,D				
CBC3	SET 0,E				
CBC4	SET 0,H				
CBC5	SET 0,L				
CBCE	SET 1,(HL)				
ddCBdCE	SET 1,(IX+d)				
FDCBdCE	SET 1,(IY+d)				
CBCF	SET 1,A				
CBC8	SET 1,B				
CBC9	SET 1,C				
CBCA	SET 1,D				
CBCB	SET 1,E				
CBCC	SET 1,H				
CBCD	SET 1,L				
CBD6	SET 2,(HL)				
DDCBdD6	SET 2,(IX+d)				
FDCBdD6	SET 2,(IY+d)				
CBD7	SET 2,A				
CBD0	SET 2,B				
CBD1	SET 2,C				
CBD2	SET 2,D				
CBD3	SET 2,E				
CBD4	SET 2,H				
CBD5	SET 2,L				
CBDE	SET 3,(HL)				
DDCBdDE	SET 3,(IX+d)				
FDCBdDE	SET 3,(IY+d)				
CBDF	SET 3,A				
CBD8	SET 3,B				
CBD9	SET 3,C				
CBDA	SET 3,D				

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
CBDB	SET 3,E				
CBDC	SET 3,H				
CBDD	SET 3,L				
CBE6	SET 4,(HL)				
DDCBdE6	SET 4,(IX+d)				
FDCBdE6	SET 4,(IY+d)				
CBE7	SET 4,A				
CBE0	SET 4,B				
CBE1	SET 4,C				
CBE2	SET 4,D				
CBE3	SET 4,E				
CBE4	SET 4,H				
CBE5	SET 4,L				
CBEE	SET 5,(HL)				
DDCBdEE	SET 5,(IX+d)				
FDCBdEE	SET 5,(IY+d)				
CBEF	SET 5,A				
CBEB	SET 5,B				
CBE9	SET 5,C				
CBEA	SET 5,D				
CBEB	SET 5,E				
CBEC	SET 5,H				
CBED	SET 5,L				
CBF6	SET 6,(HL)				
DDCBdF6	SET 6,(IX+d)				
FDCBdF6	SET 6,(IY+d)				
CBF7	SET 6,A				
CBF0	SET 6,B				
CBF1	SET 6,C				
CBF2	SET 6,D				
CBF3	SET 6,E				
CBF4	SET 6,H				
CBF5	SET 6,L				
CBFE	SET 7,(HL)				
DDCBdFE	SET 7,(IX+d)				
FDCBdFE	SET 7,(IY+d)				
CBFF	SET 7,A				
CBF8	SET 7,B				
CBF9	SET 7,C				
CBFA	SET 7,D				
CBFB	SET 7,E				
CBFC	SET 7,H				
CBFD	SET 7,L				
CB26	SLA (HL)	●	●	●	●
DDCBd26	SLA (IX+d)	●	●	●	●
FDCBd26	SLA (IY+d)	●	●	●	●
CB27	SLA A	●	●	●	●
CB20	SLA B	●	●	●	●

CODES DES INSTRUCTIONS Z80 PAR ORDRE ALPHABETIQUE

Code objet	*Instruction*	*S*	*Z*	*P/V*	*C*
CB21	SLA C	●	●	●	●
CB22	SLA D	●	●	●	●
CB23	SLA E	●	●	●	●
CB24	SLA H	●	●	●	●
CB25	SLA L	●	●	●	●
CB2E	SRA (HL)	●	●	●	●
DDCBd2E	SRA (IX+d)	●	●	●	●
FDCBd2E	SRA (IY+d)	●	●	●	●
CB2F	SRA A	●	●	●	●
CB28	SRA B	●	●	●	●
CB29	SRA C	●	●	●	●
CB2A	SRA D	●	●	●	●
CB2B	SRA E	●	●	●	●
CB2C	SRA H	●	●	●	●
CB2D	SRA L	●	●	●	●
CB3E	SRL (HL)	●	●	●	●
DDCBd3E	SRL (IX+d)	●	●	●	●
FDCBd3E	SRL (IY+d)	●	●	●	●
CB3F	SRL A	●	●	●	●
CB38	SRL B	●	●	●	●
CB39	SRL C	●	●	●	●
CB3A	SRL D	●	●	●	●
CB3B	SRL E	●	●	●	●
CB3C	SRL H	●	●	●	●
CB3D	SRL L	●	●	●	●
96	SUB (HL)	●	●	●	●
DD96d	SUB (IX+d)	●	●	●	●
FD96d	SUB (IY+d)	●	●	●	●
97	SUB A	●	●	●	●
90	SUB B	●	●	●	●
91	SUB C	●	●	●	●
92	SUB D	●	●	●	●
93	SUB E	●	●	●	●
94	SUB H	●	●	●	●
95	SUB L	●	●	●	●
D6d	SUB d	●	●	●	●
AE	XOR (HL)	●	●	●	0
DDAEd	XOR (IX+d)	●	●	●	0
FDAEd	XOR (IY+d)	●	●	●	0
AF	XOR A	●	●	●	0
A8	XOR B	●	●	●	0
A9	XOR C	●	●	●	0
AA	XOR D	●	●	●	0
AB	XOR E	●	●	●	0
AC	XOR H	●	●	●	0
AD	XOR L	●	●	●	0
EEd	XOR d	●	●	●	0

Instructions sans préfixes

- n : octet (8 bits, de 0 à 255) ;
- nn : double octet (16 bits, de 0 à 65535) ;
- d : déplacement pour l'adressage relatif (8 bits).

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø	NOP	LD BC,nn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA
1	DJNZ d	LD DE,nn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR d	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA
2	JR NZ,d	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,d	ADD HL,HL	LD HL,(nn)	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,d	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF C,d	JR C,d	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n	CCF
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A
C	RET NZ	POP BC	JP NZ,nn	JP nn	CALL NZ,nn	PUSH BC	ADD A,n	RST Ø	RET Z	RET	JP Z,nn		CALL Z,nn	CALL nn	ADC A,n	RST 8
D	RET NC	POP DE	JP NC,nn	OUT (n),A	CALL NC,nn	PUSH DE	SUB n	RST 16	RET C	EXX	JP C,nn	IN A,(n)	CALL C,nn		SBC A,n	RST 24
E	RET PO	POP HL	JP PO,nn	EX (SP),HL	CALL PO,nn	PUSH HL	AND n	RST 32	RET DE	JP (HL)	JP PE,nn	EX DE,HL	CALL PE,nn		XOR n	RST 40
F	RET P	POP AF	JP P,nn	DI	CALL P,nn	PUSH AF	OR n	RST 48	RET M	LD SP,HL	JP M,nn	EI	CALL M,nn		CP n	RST 56

Instructions avec le préfixe CB

Toutes les instructions de ce tableau doivent être précédées du préfixe CB.

	Ø	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
Ø	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC L	RRC (HL)	RRC A
1	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR L	RR (HL)	RR A
2	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA L	SRA (HL)	SRA A
3									SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
4	BIT Ø,B	BIT Ø,C	BIT Ø,D	BIT Ø,E	BIT Ø,H	BIT Ø,L	BIT Ø,(HL)	BIT Ø,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT 1,L	BIT 1,(HL)	BIT 1,A
5	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT 3,L	BIT 3,(HL)	BIT 3,A
6	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT 5,L	BIT 5,(HL)	BIT 5,A
7	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT 7,L	BIT 7,(HL)	BIT 7,A
8	RES Ø,B	RES Ø,C	RES Ø,D	RES Ø,E	RES Ø,H	RES Ø,L	RES Ø,(HL)	RES Ø,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES 1,L	RES 1,(HL)	RES 1,A
9	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES 3,L	RES 3,(HL)	RES 3,A
A	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES 5,L	RES 5,(HL)	RES 5,A
B	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES 7,L	RES 7,(HL)	RES 7,A
C	SET Ø,B	SET Ø,C	SET Ø,D	SET Ø,E	SET Ø,H	SET Ø,L	SET Ø,(HL)	SET Ø,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET 1,L	SET 1,(HL)	SET 1,A
D	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET 3,L	SET 3,(HL)	SET 3,A
E	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET 5,L	SET 5,(HL)	SET 5,A
F	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET 7,L	SET 7,(HL)	SET 7,A

Instructions avec préfixe ED

Toutes les instructions de ce tableau doivent être précédées du préfixe ED.

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø																
1																
2																
3																
4	IN B,(C)	OUT (C),B	SBC HL,BC	ID (nn),BC	NEG	RETN	IM Ø	LD I,A	IN C,(C)	OUT (C),C	ADC HL,BC	LD BC,(nn)		RETI		LD R,A
5	IN D,(C)	OUT (C),D	SBC HL,DE	LD (nn),DE			IM 1	LD A,I	IN E,(C)	OUT (C),E	ADC HL,DE	LD DE,(nn)			IM 2	LD A,R
6	IN H,(C)	OUT (C),H	SBC HL,HL	LD (nn),HL				RRD	IN L,(C)	OUT (C),L	ADC HL,HL	LD HL,(nn)				RLD
7	IN F,(C)		SBC HL,SP	LD (nn),SP					IN A,(C)	OUT (C),A	ADC HL,SP	LD SP,(nn)				
8																
9																
A	LDI	CPI	INI	OUTI					LDD	CPD	IND	OUTD				
B	LDIR	CPIR	INIR	OTIR					LDDR	CPDR	INDR	OTDR				
C																
D																
E																
F																

Instructions indexées

Toutes les instructions de ce tableau doivent être précédées par le préfixe DD, dans le cas du registre d'index IX, et de FD, pour le registre IY.

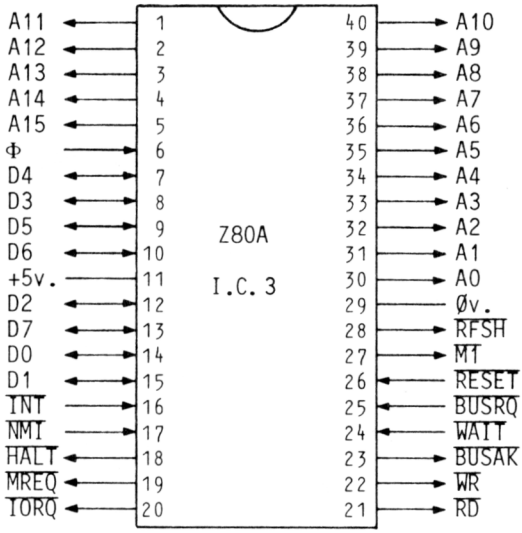
Code	*Mnémonique*	*Code*	*Mnémonique*
09	ADD IX,BC	CB d 0E	RRC (IX+d)
19	ADD IX,DE	CB d 16	RL (IX+d)
21	LD IX,nn	CB d 1E	RR (IX+d)
22	LD (nn),IX	CB d 26	SLA (IX+d)
23	INC IX	CB d 2E	SRA (IX+d)
29	ADD IX,IX	CB d 3E	SRL (IX+d)
2A	LD IX,(nn)	CB d 46	BIT 0,(IX+d)
2B	DEC IX	CB d 4E	BIT 1,(IX+d)
34	INC (IX+d)	CB d 56	BIT 2,(IX+d)
35	DEC (IX+d)	CB d 5E	BIT 3,(IX+d)
36	LD (IX+d),nn	CB d 66	BIT 4,(IX+d)
39	ADD IX,SP	CB d 6E	BIT 5,(IX+d)
46	LD B,(IX+d)	CB d 76	BIT 6,(IX+d)
4E	LD C,(IX+d)	CB d 7E	BIT 7,(IX+d)
56	LD D,(IX+d)	CB d 86	RES 0,(IX+d)
5E	LD E,(IX+d)	CB d 8E	RES 1,(IX+d)
66	LD H,(IX+d)	CB d 96	RES 2,(IX+d)
6E	LD L,(IX+d)	CB d 9E	RES 3,(IX+d)
70	LD (IX+d),B	CB d A6	RES 4,(IX+d)
71	LD (IX+d),C	CB d AE	RES 5,(IX+d)
72	LD (IX+d),D	CB d B6	RES 6,(IX+d)
73	LD (IX+d),E	CB d BE	RES 7,(IX+d)
74	LD (IX+d),H	CB d C6	SET 0,(IX+d)
75	LD (IX+d),L	CB d CE	SET 1,(IX+d)
77	LD (IX+d),A	CB d D6	SET 2,(IX+d)
7E	LD 1,(IX+d)	CB d DE	SET 3,(IX+d)
86	ADD A,(IX+d)	CB d E6	SET 4,(IX+d)
8E	ADC A,(IX+d)	CB d EE	SET 5,(IX+d)
96	SUB (IX+d)	CB d F6	SET 6,(IX+d)
9E	SBC A,(IX+d)	CB d FE	SET 7,(IX+d)
A6	AND (IX+d)	E1	POP IX
AE	XOR (IX+d)	E3	EX (SP),IX
B6	OR (IX+d)	E5	PUSH IX
BE	CP (IX+d)	E9	JP (IX)
CB d 06	RLC (IX+d)	F9	LD SP,IX

CONNECTEUR ARRIERE DU ZX-81

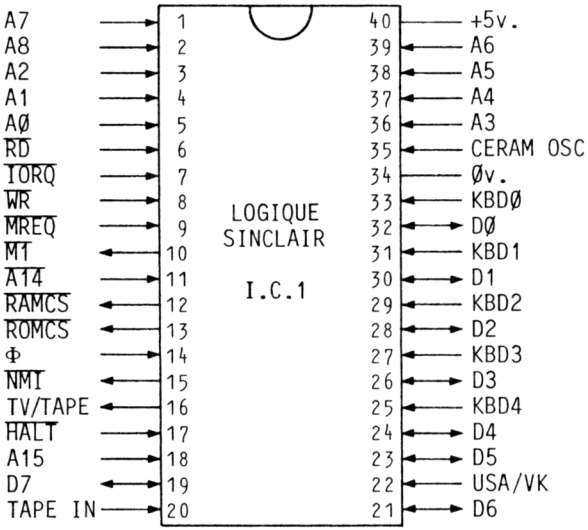
Coté soudure				*Coté composants*
5 volts régulés	5v.	1	D7	Bit 7 de donnée
9 volts semi-régulés	9v.	2	$\overline{RAM}$	Commande RAM
		détrompeur		
Ø volts	Øv.	4	DØ	Bit Ø de donnée
Ø volts	Øv.	5	D1	Bit 1 de donnée
Horloge	$\overline{\Phi}$	6	D2	Bit 2 de donnée
Bit Ø d'adresse	A0	7	D6	Bit 6 de donnée
Bit 1 d'adresse	A1	8	D5	Bit 5 de donnée
Bit 2 d'adresse	A2	9	D3	Bit 3 de donnée
Bit 3 d'adresse	A3	10	D4	Bit 4 de donnée
Bit 15 d'adresse	A15	11	$\overline{INT}$	Interruptions
Bit 14 d'adresse	A14	12	$\overline{NAY}$	Interruptions masquables
Bit 13 d'adresse	A13	13	$\overline{HALT}$	Signal d'arret du microprocesseur
Bit 12 d'adresse	A12	14	$\overline{MREQ}$	Opération mémoire
Bit 11 d'adresse	A11	15	$\overline{I/O\,RQ}$	Opération d'entrée/sortie
Bit 10 d'adresse	A10	16	$\overline{RD}$	Commande de lecture
Bit 9 d'adresse	A9	17	$\overline{WR}$	Commande d'écriture
Bit 8 d'adresse	A8	18	$\overline{BUSAK}$	Acceptation d'accès direct mémoire
Bit 7 d'adresse	A7	19	$\overline{WAIT}$	Signal d'attente du microprocesseur
Bit 6 d'adresse	A6	20	$\overline{BUSRQ}$	Demande d'accès direct mémoire
Bit 5 d'adresse	A5	21	$\overline{RESET}$	Mise à zéro générale
Bit 4 d'adresse	A4	22	$\overline{MT}$	Signal de début de cycle
Commande ROM	$\overline{ROM}$	23	$\overline{REFSH}$	Commande de rafraichissement des mémoires dynamiques

Remarque : les signaux surmontés d'une barre sont vrais à l'état bas (Ø volt).

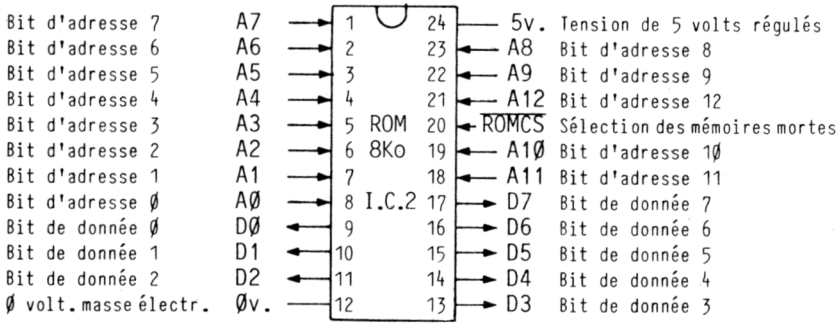
BROCHAGE DES CIRCUITS INTEGRES



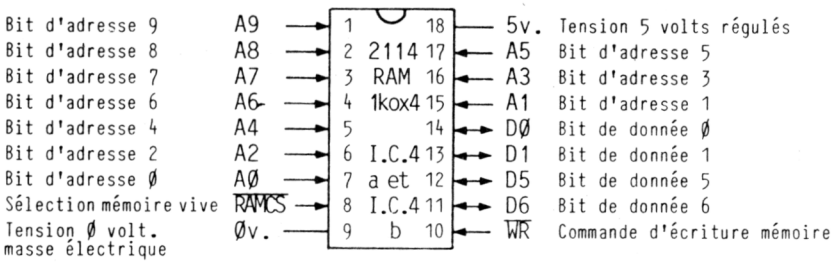
1-Bit d'adresse 11
2-Bit d'adresse 12
3-Bit d'adresse 13
4-Bit d'adresse 14
5-Bit d'adresse 15
6-Horloge
7-Bit de donnée 4
8-Bit de donnée 3
9-Bit de donnée 5
10-Bit de donnée 6
11-Tension de 5 volts régulés
12-Bit de donnée 2
13-Bit de donnée 7
14-Bit de donnée Ø
15-Bit de donnée 1
16-Interruptions masquables
17-Interruptions non masquables
18-Signal d'arrêt du microprocesseur
19-Demande d'opération mémoire
20-Demande d'entrées/sorties
21-Commande de lecture mémoire
22-Commande d'écriture mémoire
23-Acceptation d'accès direct mémoire
24-Demande d'attente au microprocesseur
25-Demande d'accès direct mémoire
26-Initialisation du microprocesseur
27-Signal de début de cycle
28-Rafraichissement des mémoires dynamiques
29-Tension Ø volt masse électrique
30-Bit d'adresse Ø
31-Bit d'adresse 1
32-Bit d'adresse 2
33-Bit d'adresse 3
34-Bit d'adresse 4
35-Bit d'adresse 5
36-Bit d'adresse 6
37-Bit d'adresse 7
38-Bit d'adresse 8
39-Bit d'adresse 9
40-Bit d'adresse 10



1-Bit d'adresse 7
2-Bit d'adresse 8
3-Bit d'adresse 2
4-Bit d'adresse 1
5-Bit d'adresse Ø
6-Commande de lecture mémoire
7-Demande d'entrées/sorties
8-Commande d'écriture mémoire
9-Demande d'opération mémoire
10-Signal de début de cycle
11-Bit d'adresse 14
12-Sélection des circuits mémoire vive
13-Sélection des circuits mémoire morte
14-Horloge quartz
15-Interruption non masquable
16-Signal télévision et micro
17-Signal d'arrêt du microprocesseur
18-Bit d'adresse 15
19-Bit de donnée 7
20-Signal du magnéto K7
21-Bit de donnée 6
22-A la masse dans la version USA
23-Bit de donnée 5
24-Bit de donnée 4
25-Bit de donnée 4 du clavier (colonne)
26-Bit de donnée 3
27-Bit de donnée 3 du clavier (colonne)
28-Bit de donnée 2
29-Bit de donnée 2 du clavier (colonne)
30-Bit de donnée 1
31-Bit de donnée 1 du clavier (colonne)
32-Bit de donnée Ø
33-Bit de donnée Ø du clavier (colonne)
34-Tension Ø volt masse électrique
35-Oscillateur céramique
36-Bit d'adresse 3
37-Bit d'adresse 4
38-Bit d'adresse 5
39-Bit d'adresse 6
40-Tension 5 volts régulés

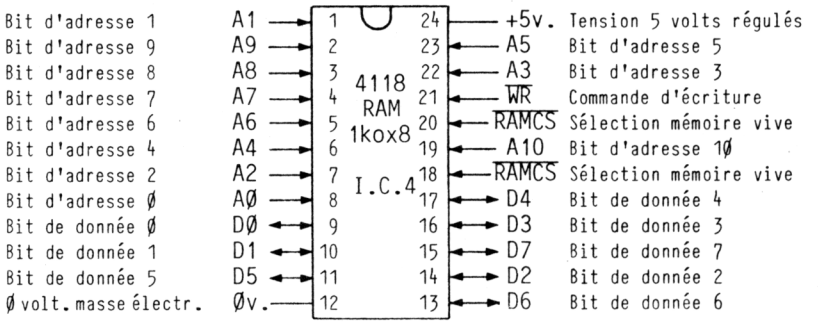


Mémoire vive, première version : 2 circuits 2114

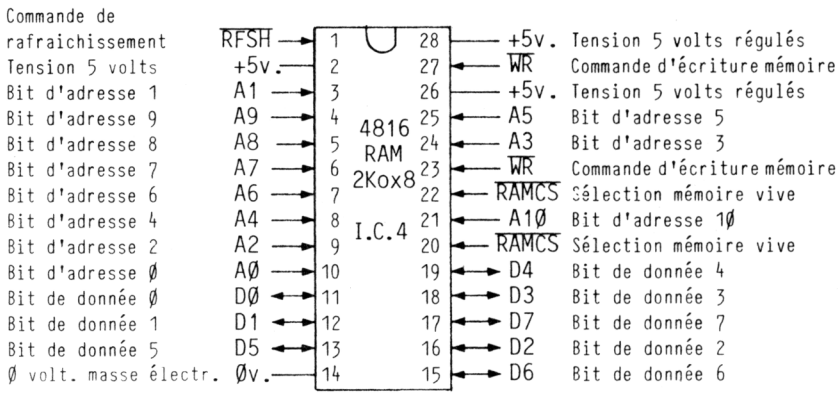


Broches	*I.C.4 a*	*I.C.4 b*
11	D2	D6
12	D7	D5
13	D3	D1
14	D4	DØ

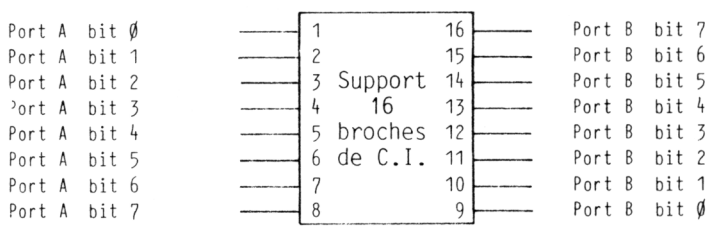
Mémoire vive, deuxième version : 1 circuit 4118



Mémoire vive, troisième version : 1 circuit 4816 (TIMEX 1000)



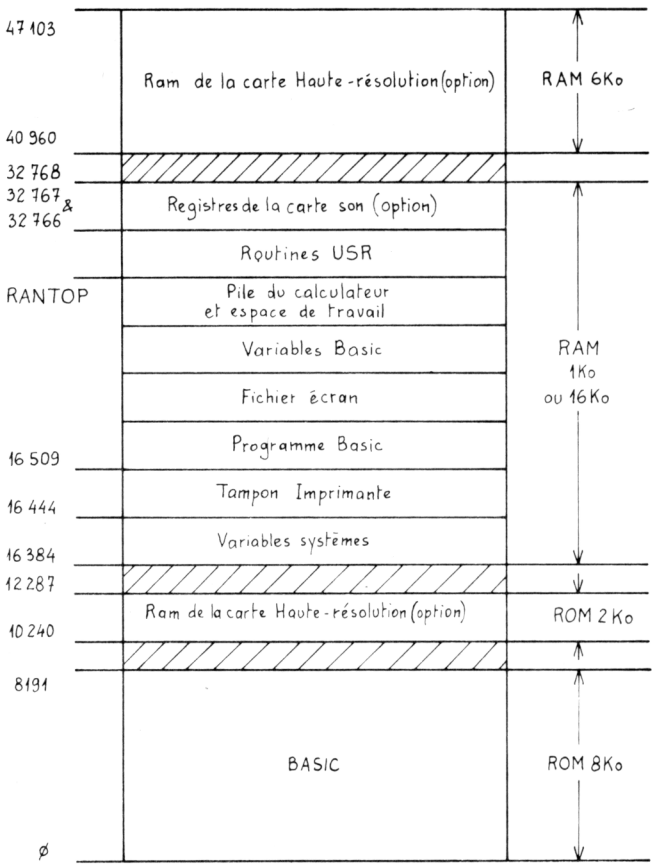
Ports d'entrées/sorties de la carte Son QS



(voir chapitre "Adresses").

CARTE MEMOIRES

Les parties hachurées sont des emplacements libres dans lesquels il est possible d'insérer de la mémoire vive ou des EPROMs (mémoire morte programmable électriquement), comme le fait la carte haute-résolution.



ADRESSES DE LA ROM BASIC (de 0000 à 1DFF)

Ces emplacements mémoires sont des points d'entrée utilisables par vos programmes en langage machine.

Routine RST

00 Initialisation. C'est la première instruction exécutée par le microprocesseur après un Reset. La routine vérifie la mémoire disponible et place dans RAMTOP l'adresse du dernier octet + 1.

08 Routine d'erreur. L'octet qui suit RST08 donne le type d'erreur. Le programme retourne automatiquement au Basic.

10 Routine d'affichage d'un caractère. Ce dernier doit être placé dans l'accumulateur avant RST10. Il est affiché à la position donnée par la variable système DF-CC (adresse 400E).

18 Donne dans l'accumulateur le prochain caractère à interpréter par le Basic. Ce caractère est pointé par la variable système CH-ADD.

20 Identique à RST18, mais pour le caractère suivant.

28 Routine du calculateur en virgule flottante. Les opérandes sont placées dans la pile du calculateur pointée par la variable système STKBOT.

30 Réserve de la place pour les variables. Le registre BC doit contenir le nombre d'octets que l'on veut ainsi récupérer.

38 Routine d'interruption masquable. Cette routine génère une ligne sur l'écran à chaque interruption.

66 Routine d'interruption non masquable. Cette routine est exécutée à chaque trame de télévision en mode SLOW.

007E Table des caractères du clavier.

00CC Table des caractères de fonctions.

00F3 Table des caractères graphiques.

0111 Table des caractères de commandes.

0207 Routine d'affichage.

02F6 Routine de la commande SAVE.

0340 Routine de la commande LOAD.

03C3 Routine de la commande NEW.

03CB Routine d'initialisation.

0419 Routine d'édition d'une ligne Basic.

0454 Routine de déplacement du curseur vers le bas.

0482 Routine de gestion de la ligne en cours d'entrée (E-LINE).

04C1 Point d'entrée des routines de commande.

052B Routine déterminant les touches éditées.

05C4 Routine d'édition.

063E Routine d'exécution d'un programme Basic.

072C Routine de la commande LLIST.

0730 Routine de la commande LIST.

0745 Routine d'affichage d'une ligne Basic complète.

07BD Routine de décodage du clavier.

07F1 Routine d'affichage d'un caractère.

0869 Routine de la commande COPY.

08F5 Routine de test des paramètres de la commande PRINT AT.

0918 Routine qui agrandit le fichier d'affichage.

094B Routine d'affichage des mots réservés.

09F2 Routine qui détermine la prochaine variable ou la prochaine ligne de Basic.

0A2A Routine de la commande CLS.

0A98 Routine d'affichage d'un nombre décimal.

0ACB Routine de la commande LPRINT.

0ACF Routine de la commande PRINT.

0BAF Routine des commandes PLOT et UNPLOT.

0C0E Routine de la commande SCROLL.

Table des commandes

	Commandes	*Adresse de la routine*
0C48	**LET**	131D
0C4B	**GOTO**	0E81
0C4F	**IF**	0DAB
0C54	**GOSUB**	0EB5
0C58	**STOP**	0CDC
0C5B	**RETURN**	0ED8
0C5E	**FOR**	0DB9
0C66	**NEXT**	0E2E
0C6A	**PRINT**	0ACF

→

ADRESSES DE LA ROM BASIC
(de 1000 à 1DFF)

	Commandes	*Adresse de la routine*
0C6D	**INPUT**	0EE9
0C71	**DIM**	1405
0C74	**REM**	0D6A
0C77	**NEW**	03C3
0C7A	**RUN**	0EAF
0C7D	**LIST**	0730
0C80	**POKE**	0E92
0C86	**RAND**	0E6C
0C89	**LOAD**	0340
0C8C	**SAVE**	02F6
0C8F	**CONT**	0E7C
0C92	**CLEAR**	1496
0C95	**CLS**	0A2A
0C98	**PLOT**	0BAF
0C9E	**UNPLOT**	0BAF
0CA2	**SCROLL**	0C0E
0CA7	**PAUSE**	0F2F
0CAB	**SLOW**	0F28
0CAE	**FAST**	0F20
0CB1	**COPY**	0869
0CB4	**LPRINT**	0ACB
0CB7	**LLIST**	072C

0CDC Routine de la commande STOP
0D6A Routine de la commande REM.
0DAB Routine de la commande IF.
0DB9 Routine de la commande FOR.
0E2E Routine de la commande NEXT.
0E6C Routine de la commande RAND.
0E7C Routine de la commande CONT.
0E81 Routine de la commande GOTO.
0E92 Routine de la commande POKE.
0EB5 Routine de la commande GOSUB.
0ED8 Routine de la commande RETURN.
0EE9 Routine de la commande INPUT.
0EAF Routine de la commande RUN.
0F23 Routine de la commande FAST.
0F2B Routine de la commande SLOW.

0F32 Routine de la commande PAUSE.

0F55 Routine d'évaluation d'une expression numérique ou d'une chaîne de caractères. Le bit 6 de la variable system FLAGS est mis à 1 si le résultat est numérique.

110F Table des priorités.

111C Cherche de la place pour stocker les variables.

11A7 Donne dans HL l'emplacement des caractères d'une chaîne de caractères.

1263 Découpe une chaîne de caractères dont les paramètres se trouvent en haut de la pile du calculateur.

12C2 Place dans la pile du calculateur le contenu des registres A, B, C, D et E.

12FF Place dans le registre DE le contenu de l'adresse pointée par DE+1.

1305 Place dans le registre HL le résultat de l'opération HL*DE.

131D Routine de la commande LET.

13F8 Place dans la pile STKEND les cinq octets d'un nombre en virgule flottante ou les paramètres d'une chaîne de caractères provenant de la pile du calculateur.

1409 Routine de la commande DIM.

1488 Réserve de la mémoire pour l'espace de travail (E-LINE).

149A Routine de la commande CLEAR.

14AD Place dans l'espace de travail le curseur suivi d'un NEWLINE pour l'entrée des lignes de programme.

14CE Met à un le drapeau de retenue si le registre A contient une lettre de l'alphabet.

14D2 Met à un le drapeau de retenue si le registre A contient une lettre ou un chiffre.

14D9 Convertit un nombre décimal d'une ligne de Basic en nombre en virgule flottante utilisable par le calculateur. Le résultat est placé en haut de la pile du calculateur.

1548 Convertit un nombre entier d'une ligne de Basic (ou un numéro de ligne) au format virgule flottante.

155A Convertit un nombre sous forme exponentielle au format virgule flottante.

158A Compresse le nombre en virgule flottante de la pile du calculateur dans le registre BC. Si le résultat est trop grand (supérieur à 65535), le drapeau de retenue est mis à 1.

15DB Affiche à l'écran le nombre en virgule flottante de la pile du calculateur (utilisé par PRINT et STR$).

16D8 Prépare aux routines d'addition, soustraction, multiplication et division en effectuant un complément à 2 pour les nombres négatifs. L'exposant est placé dans le registre A et le premier octet est 0 pour les nombres positifs et FF pour les négatifs.

16F7 Place les deux opérandes de la pile du calculateur pour l'addition, la multiplication et la soustraction dans les registres. HL doit pointer le premier octet du premier nombre et DE celui du second.

174C Effectue une soustraction en changeant le signe de la deuxième opérande avant d'appeler la routine d'addition.

1755 Additionne deux nombres en virgule flottante.

17C6 Multiplie deux nombres en virgule flottante.

1882 Divise deux nombres en virgule flottante.

18E4 Tronque un nombre pour tendre vers 0 (5.2 donne 5 et -5.2 donne -5).

Table des constantes utilisées par le calculateur au format virgule flottante

1915 0
1918 1
191A 1/2
191C 1/2 de PI
1921 10

Table des adresses des fonctions utilisant les routines en virgule flottante

Ces routines sont appelées par la fonction RST28. Le nombre qui suit cette instruction donne le type d'opération à effectuer. Une liste d'opérations doit toujours se terminer par le code 34 (fin des calculs).

	Routine	*Octet après RST28*	*Adresse routine*
1923	Saut si exact	00	1C2F
1925	Echange les opérandes	01	1A72
1927	Suppression	02	19E3
1929	Soustraction	03	174C
192B	Multiplication	04	17C6

	Routine	*Octet après RST28*	*Adresse routine*
192D	Division	05	1882
192F	Puissance	06	1DE2
1931	Or	07	1AED
1933	And	08	1AF3
1935		09	1B03
1937		0A	1B03
1939		0B	1B03
193B		0C	1B03
193D		0D	1B03
193F		0E	1B03
1941	Addition	0F	1755
1943	AND entre chaîne et numér.	10	1AF8
1945		11	1B03
1947		12	1B03
1949		13	1B03
194B		14	1B03
194D		15	1B03
194F		16	1B03
1951		17	1B62
1953	Négation	18	1AA0
1955	CODE	19	1C06
1957	VAL	1A	1BA4
1959	LEN	1B	1C11
195B	SIN	1C	1D49
195D	COS	1D	1D3E
195F	TAN	1E	1D6E
1961	ASN	1F	1DC4
1963	ACS	20	1DD4
1965	ATN	21	1976
1967	LN	22	1CA9
1969	EXP	23	1C5B
196B	INT	24	1C46
196D	SQR	25	1DDB
196F	SGN	26	1AAF
1971	ABS	27	1AAA
1973	PEEK	28	1ABE
1975	USR	29	1AC5
1977	STR$	2A	1BD5
1979	CHR$	2B	1B8F
197B	NOT	2C	1AD5
197D	Déplace un nombre en v.f.	2D	19F6
197F	Modulo	2E	1C37
1981		2F	1C23
1983		30	19FC
1985	Décrément et saut si	31	1C17
1987	non Ø < 0	32	1ADB

	Routine	*Octet après RST28*	*Adresse routine*
1989	> 0	33	1ACE
198B	Fin des calculs	34	002B
198D	Réduit l'argument	35	1D18
198F	Tronque un nombre	36	18E4
1991		37	19E4
1993	Conversion exponentielle	38	155A
1995		39	1A7F
1997		3A	1A51
1999		3B	1A63
199B		3C	1A45

199D Routine du calculateur.

19E3 Routine de suppression d'une opérande de la pile du calculateur.

19E4 Utilisée par 0F55 pour les opérations simples.

19EB Test s'il reste au moins cinq octets de libres dans la pile du calculateur pour stocker un nombre en virgule flottante.

19F6 Déplace un nombre en virgule flottante en haut de la pile du calculateur.

1A2D Routine de recherche d'une constante du calculateur. HL doit contenir l'adresse de la table et A le numéro de la constante. Au retour, HL contient l'adresse de la constante.

1A3C Cherche l'adresse des blocs de cinq octets contenus dans la pile du calculateur.

1A45 Déplace un nombre en virgule flottante de la zone mémoire du calculateur (variable MEM) dans la pile du calculateur.

1A51 Place une constante (table en 1915) dans la pile du calculateur.

1A63 Déplace un nombre en virgule flottante de la pile du calculateur dans la zone mémoire (variable MEM).

1A72 Echange le contenu des deux nombres en virgule flottante placés en haut de la pile du calculateur.

1A7F Génère la série de nombres utilisés pour le calcul des SIN, ATN, LN et EXP ainsi que leur dérivés (COS, TAN, ASN, ACS, ** et SQR).

1AA0 Change le signe de la dernière valeur de la pile du calculateur.

1AAA Routine de la fonction ABS.

1AAF Routine de la fonction SGN.

1ABE Routine de la fonction PEEK.

1AC5 Routine de la fonction USR.

1ACE Donne la valeur 1 à la dernière valeur de la pile si la valeur de celle pointée par HL est supérieure à 0, et 0 dans les autres cas.

1AD5 Routine de la fonction NOT.

1ADB Donne la valeur 1 à la dernière valeur de la pile si la valeur de celle pointée par HL est inférieure à 0, et 0 dans les autres cas.

1AE0 Donne la valeur 1 à la dernière valeur de la pile si le drapeau de retenue est à 1, et 0 s'il est à 0.

1AED Routine de la fonction OR.

1AF3 Routine de la fonction AND entre deux valeurs numériques.

1AF8 Routine de la fonction AND entre une chaîne de caractères et une valeur numérique.

1B62 Concatène deux chaînes de caractères.

1B85 Place, dans HL, l'adresse du premier octet de la dernière valeur de la pile (STKEND-5) et, dans DE, celle du dernier octet (STKEND).

1B8F Routine de la fonction CHR$.

1BA4 Routine de la fonction VAL.

1BD5 Routine de la fonction STR$.

1C06 Routine de la fonction CODE.

1C11 Routine de la fonction LEN.

1C17 Effectue une fonction machine DJNZ, mais avec la variable système BERG.

1C46 Routine de la fonction INT.

1C5B Routine de la fonction EXP.

1CA9 Routine de la fonction LN.

1D3E Routine de la fonction COS.

1D49 Routine de la fonction SIN.

1D6E Routine de la fonction TAN.

1D76 Routine de la fonction ATN.

1DC4 Routine de la fonction ASN.

1DD4 Routine de la fonction ACS.

1DDB Routine de la fonction SQR.

1DE2 Routine de la fonction **.

ADRESSE DE L'EPROM DE LA CARTE HAUTE-RESOLUTION (OPTION)

POINTS D'ENTREE DES FONCTIONS H-R : de 2800 à 2FFF

2800 Interpréteur de ligne.

2803 Routine de test de la carte H-R.

2806 Tracé en noir sur fond blanc.

2809 Tracé en blanc sur fonc noir.

280C Efface l'écran H-R.

280F Déplace le curseur aux coordonnées X,Y. Le registre HL doit contenir ces coordonnées où H=Y et L=X.

2812 Ecrit un point aux coordonnées X,Y. Le registre HL doit contenir ces coordonnées où H=Y et L=X.

2815 Trace une ligne de la position actuelle aux coordonnées X,Y contenues dans HL.

2818 Affiche la chaîne A$. En langage machine, HL doit contenir l'adresse du premier octet de la chaîne et l'accumulateur sa longueur.

281B Déplace l'écran vers le haut pour laisser trois lignes pour les messages.

281E Déplace le contenu de l'écran vers le haut d'une ligne de pixels.

2821 Identique à 281E, mais vers le bas.

2824 Déplace le contenu de l'écran vers la gauche d'une colonne de pixels.

2827 Identique à 2824, mais vers la droite.

282A Recopie le contenu de l'écran H-R sur l'imprimante.

282D Dessine une boite de la position actuelle du curseur aux coordonnées placées dans le registre HL.

Remarque : les coordonnées X et Y sont placées dans la variable système COORDS (4036) et la couleur à l'adresse 407B (inutilisée par le ZX-81).

Ces variables sont utilisées en permanence par le Basic. Elles peuvent être modifiées par vos programmes en langage machine avec cependant quelques réserves :

0 : peut être modifiée par vos programmes en Basic ou L.M.
N : ne peut pas être modifiée ("plante" votre machine).
RIEN : le ZX-81 change cette valeur immédiatement.

Adresse	Nombre d'octets	Modification	Nom de la variable	
4000	1	0	ERR-NR	Code de l'erreur rencontrée. Cette valeur correspond à 1 moins le code erreur. S'il n'y a pas d'erreur, on trouve dans cette variable 255d. Il est possible de l'utiliser pour vos propres codes d'erreur avec des valeurs entre 15d et 34d (lettres de G à Z, car il faut ajouter 29d) en utilisant la routine RST08 (voir RST08 dans la ROM).
4001	1	N	FLAGS	Cette variable contient les différents drapeaux utilisés par les routines d'évaluation (0F55), de recherche de place mémoire (111C), d'emplacement d'une chaîne (11A7), de découpe de paramètres (1263), de déplacement des registres (12C2), de la fonction LET (1321) ou de la fonction VAL (1BA4).
4002	2	N	ERR-SP	Adresse de la pile des instructions GOSUB.
4004	2	0	RAMTOP	Adresse du premier octet non utilisé par le Basic. Cette variable peut être modifiée pour réserver de la place pour les programmes en langage machine par l'instruction POKE suivie de la commande NEW ou CLS. Sans une de ces deux dernières commandes, l'instruction POKE n'a aucun effet sur RAMTOP.
4006	1		MODE	Type de curseur sur l'écran (K, L, F ou G).
4007	2		PPC	Numéro de la ligne en cours d'exécution.
4009	1	0	VERSN	Identification du ZX-81 dans la sauvegarde d'un programme.
400A	2	0	E-PPC	Numéro de la ligne en cours d'exécution et du pointeur du caractère dans cette ligne.
400C	2	N	D-FILE	Pointeur de l'adresse de début du fichier d'affichage.
400E	2	0	DF-CC	Adresse de la prochaine position d'écriture par l'instruction PRINT. Cette variable système est utilisée par RST10 et la routine de la fonction STR$.

Adresse	Nombre d'octets	Modification	Nom de la variable	
4010	2	N	VARS	Pointeur du début de la table des variables utilisées par le Basic. Cette variable est employée par la routine de recherche de place mémoire (111C) ou par la commande CLEAR (149A).
4012	2		DEST	Adresse de la variable en cours d'affectation. Cette variable est utilisée par la routine de la commande LET (1321).
4014	2	N	E-LINE	Pointeur du tampon de la ligne Basic en cours d'entrée et de l'espace de travail. Cette variable est utilisée par les routines de gestion de la ligne en cours (0482), de l'instruction LET (1321), de l'instruction DIM (1409), de réservation mémoire (1488), de la commande CLEAR (149A) ou d'initialisation de l'espace de travail (14AD).
4016	2	N	CH-ADD	Adresse du prochain caractère à interpréter. Cette variable est utilisée par les routines d'évaluation (0F55), des paramètres d'une chaîne (11A7) ou de la fonction VAL (1BA4).
4018	2	0	X-PTR	Adresse du caractère précédant le symbole d'erreur de syntaxe dans l'entrée d'une ligne Basic ou dans l'exécution de la commande INPUT.
401A	2	N	STKBOT	Pointeur de l'adresse du début de la pile du calculateur utilisée pour les calculs des nombres en virgules flottantes. Cette variable est utilisée par les routines de l'instruction LET, de réservation mémoire (1488), de l'instruction CLEAR (149A) ou d'initialisation de l'espace de travail (14AD).
401C	2	N	STKEND	Pointeur de l'adresse de fin de pile du calculateur. Cette variable est utilisée par les routines d'évaluation (0F55), de sauvegarde des registres (12C2), de sauvegarde de paramètres (13F8), de l'instruction CLEAR (149A) ou par le calculateur (199D).

Adresse	Nombre d'octets	Modification	Nom de la variable	
401E	1		BERG	Sauvegarde du registre B par le calculateur. Cette variable est utilisée dans les routines du compteur (1C17), du calculateur (199D) ou d'opération simple (19E4).
401F	2		MEM	Adresse de la pile spéciale du calculateur (généralement égal à MEMBOT). Cette variable est utilisée par les routines d'initialisation de l'espace de travail (14AD) ou de déplacement de nombres en virgule flottante (1A45 et 1A63).
4021	1	0	----	Inutilisée par le Basic (vous pouvez l'utiliser dans vos programmes en langage machine).
4022	1	N	DF-SZ	Nombre de lignes vides en dessous de l'écran + 1.
4023	2	0	S-TOP	Numéro de la ligne la plus haute dans une liste automatique.
4025	2		LAST-K	Code de la dernière touche enfoncée (voir liste des codes clavier).
4027	1		REBOND	Etat de l'anti-rebond du clavier.
4028	1		MARGIN	Nombre de lignes autour de l'image du téléviseur (55 pour la France).
4029	2	N	NXTLIN	Adresse de la ligne suivante à exécuter.
402B	2	0	OLDPPC	Numéro de la ligne à laquelle l'instruction CONT doit repartir.
402D	1		FLAGX	Drapeaux utilisés par la routine de l'instruction LET (1321).
402E	2		STRLEN	Longueur de la chaîne de caractères à interpréter. Utilisée par la routine de LET (1321).
4030	2		T-ADDR	Adresse de la prochaine rubrique dans la table de syntaxe.
4032	2	0	SEED	Base de la série des nombres aléatoires (RND). Cette variable système est déterminée par l'instruction RAND.

Adresse	Nombre d'octets	Modification	Nom de la variable	
4034	2	0	FRAMES	Compteur de trames du téléviseur. Les bits 0 à 14 de cette variable sont décrémentés à chaque trame. L'instruction PAUSE met le bit 15 à 1 et sa durée dans les autres bits. Dès que le temps est écoulé ou qu'une touche est actionnée, le bit 15 est remis à 0.
4036	1	0	COORDS	Coordonnée X du point à afficher sur l'écran par l'instruction PLOT. Cette variable est aussi utilisée par les instructions de la carte haute-résolution.
4037	1	0	COORDS	Identique à 4036, mais pour la coordonnée Y.
4038	1	0	PR-CC	Pointeur du prochain caractère à imprimer dans le tampon BRBUFF.
4039	1	N	S-POSN	Numéro de la colonne de la prochaine instruction PRINT. Utilisée aussi par la fonction STR$ (1BD5).
403A	1	N	S-POSN	Numéro de la ligne de la prochaine instruction PRINT.
403B	1	0	CDFLAG	Drapeaux du Basic.
403C	33	0	BRBUFF	Tampon de l'imprimante. Tous les caractères de la prochaine ligne à imprimer doivent être placés dans cette zone mémoire. Le dernier caractère doit être un NEWLINE (code 118d).
405D	30		MEMBOT	Zone de travail pour les nombres utilisés par le calculateur et qu'il n'est pas possible de placer dans sa pile. Cette variable est utilisée par la routine d'initialisation de l'espace de travail (14AD) ou de conversion d'un nombre décimal (14D9).
407B	2	0	------	Doubles octets inutilisés par le Basic. Cette variable est cependant utilisée par la carte haute-résolution pour y placer la couleur en cours.

7FFF - Sélection d'un registre

La valeur, placée à cette adresse, sélectionne un des registres suivants :

SELECTION		DONNEES	
7FFF	Rôle du registre	Valeur dans 7FFE	Rôle de la donnée
Ø 1	Ton aigu canal A (8bits) Ton grave canal A (4bits)	Ø à 255 Ø à 15	Contrôle fréquences
2 3	Ton aigu canal B (8bits) Ton grave canal B (4bits)	Ø à 255 Ø à 15	Contrôle fréquences
4 5	Ton aigu canal C (8bits) Ton grave canal C (4bits)	Ø à 255 Ø à 15	Contrôle fréquences
6	Bruitage	Ø à 31	
7	Commutation	Ø à 255	Voir tableau ci-dessous
8 9 10	Volume canal A Volume canal B Volume canal C	Ø à 16 Ø à 16 Ø à 16	Volume croissant de Ø à 15. 16 déclenche l'enveloppe.
11 12	Fréquence aiguë Fréquence grave	Ø à 255 Ø à 255	Détermine la durée de l'enveloppe
13	Contrôle de l'enveloppe	Ø à 15	Voir tableau ci-dessous
14 15	Port A d'E/S (8bits) Port B d'E/S (8bits)	Ø à 255 Ø à 255	

7FFE - Registre 7 (communication)

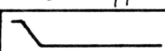
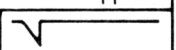
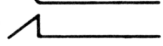
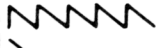
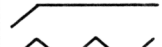
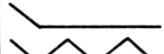
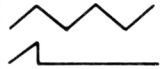
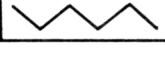
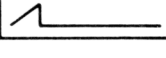
Le tableau suivant donne les valeurs à placer dans le registre de données (après sélection du registre 7) pour brancher un des canaux sons ou un des ports d'entrées/sorties.

Bit	Bit Ø	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Contrôle de	Fréquence / tonalité			Enveloppe / bruit			Entrées/sorties	
Port au canal	Canal A	Canal B	Canal C	Canal A	Canal B	Canal C	Port A	Port B
Valeur	1	2	4	8	16	32	64	128

Dans le cas où d'autres valeurs ont déjà été données, et pour ne pas les modifier, il faut suivre le tableau suivant :

	Fréquences canal			Bruit canal			E/S port	
	A	B	C	A	B	C	A	B
Pour brancher	-1	-2	-4	-8	-16	-32	-64	-128
Pour débrancher	+1	+2	+4	+8	+16	+32	+64	+128

Le contrôle de l'enveloppe est basé sur ce tableau (il est préférable d'effectuer des essais pour mieux apprécier leurs effets).

Binaire	Décimal	Forme de l'enveloppe	Binaire	Décimal	Forme de l'enveloppe
ØØXX	Ø à 3		1011	11	
01XX	4 à 7		1100	12	
1000	8		1101	13	
1001	9		1110	14	
1010	10		1111	15	

Chaque note est codée sur 12 bits. Il faut donc placer les valeurs du tableau suivant en deux fois, en sélectionnant d'abord le ton grave puis le ton aigu d'un des trois canaux A, B ou C.

Note	Octave 1 Décimal	Hex	Octave 2 Décimal	Hex	Octave 3 Décimal	Hex	Octave 4 Décimal	Hex	Octave 5 Décimal	Hex
Do	13,093	D,5D	06,174	6,AE	03,087	3,57	01,172	1,AC	00,214	0,D6
Do#	12,156	C,9C	06,078	6,4E	03,039	3,27	01,148	1,94	00,202	0,CA
Ré	11,231	B,E7	05,244	5,F4	02,250	2,FA	01,125	1,7D	00,190	0,BE
Ré#	11,060	B,3C	05,158	5,9E	02,207	2,CF	01,104	1,68	00,180	0,B4
Mi	10,155	A,9B	05,077	5,4D	02,167	2,A7	01,083	1,53	00,170	0,AA
Fa	10,002	A,02	05,001	5,01	02,129	2,81	01,064	1,40	00,160	0,A0
Fa#	09,115	9,73	04,185	4,B9	02,093	2,5D	01,046	1,2E	00,151	0,97
Sol	08,235	8,EB	04,117	4,75	02,059	2,3B	01,029	1,1D	00,143	0,8F
Sol#	08,107	8,6B	04,053	4,35	02,027	2,1B	01,013	1,0D	00,135	0,87
La	07,242	7,F2	03,249	3,F9	01,252	1,FC	00,254	0,FE	00,127	0,7F
La#	07,128	7,80	03,192	3,C0	01,224	1,E0	00,240	0,F0	00,120	0,78
Si	07,020	7,14	03,138	3,8A	01,197	1,C5	00,226	0,E2	00,113	0,71

ECRAN DE 24 LIGNES

Pour écrire sur les deux lignes interdites du bas de l'écran (réservé pour les INPUTs et les éditions de lignes Basic), il suffit de faire POKE 16442,4. Les caractères du prochain PRINT seront automatiquement écrits sur les deux dernières lignes. L'adresse mémoire 16442 correspond à la variable système S-POSN (numéro de ligne du prochain PRINT).

AFFICHAGE DIRECT ECRAN

La variable système DF-CC nous donne l'adresse du premier caractère du fichier d'affichage. Il suffit donc de "POKER" des caractères à partir de cette adresse pour les voir apparaître sur l'écran. Chaque ligne d'écran est composée de 32 caractères suivis d'un NEWLINE (code 118) pour les ZX-81 comportant plus de 3,5 kilo-octets. Il faut donc faire attention à ne pas détruire le 33e caractère, sous peine de "planter" votre machine. Pour obtenir un adressage direct, il faut écrire les lignes suivantes :

```
10 E=PEEK 16398+256*PEEK 16399
20 POKE E+(Y*33)+X,C
```

où E représente l'adresse du premier caractère du fichier d'affichage, X et Y l'adresse sur l'écran (de 0 à 31 et de 0 à 21 respectivement) et C le code du caractère à écrire. Cette méthode a l'avantage d'être beaucoup plus rapide qu'un PRINT AT.

EXECUTION AUTOMATIQUE APRES CHARGEMENT

L'instruction SAVE du ZX-81, utilisée dans un programme, sauve sur cassette non seulement les lignes Basic, mais aussi les variables et les tableaux, les variables système, le fichier d'affichage et le numéro de la prochaine ligne à exécuter. Si l'on place un GOTO n° ligne comme prochaine instruction à exécuter, le programme ne vous donnera pas la main après un LOAD, mais effectuera automatiquement cette instruction. C'est aussi un moyen de protéger vos programmes si vous faites appel à des sous-programmes en langage machine (la touche BREAK devenant inopérante).

Exemple

```
       10 REM DEBUT
       20 ...
          ...
     1000 SAVE "PROGRAMME"
     1010 GOTO 10
```

Au prochain chargement par LOAD, le programme exécute directement la ligne 10.

CHAINAGE DE PROGRAMMES

En utilisant l'astuce n° 3, il est possible de chaîner plusieurs programmes (découpage d'un programme en plusieurs parties s'il ne tient pas entièrement en mémoire). Il suffit de sauver tous les morceaux de programme par l'astuce n° 3 et d'ajouter des instructions LOAD "PROGRAMME SUIVANT" pour charger et exécuter automatiquement la suite. Exemple :

```
Programme 1
       10 REM DEBUT
       20 PRINT "EXECUTION PREMIERE PARTIE"
       30 ...
          ...
     1000 LOAD "PROGRAMME 2"

Programme 2
       10 REM DEBUT
       20 PRINT "EXECUTION DEUXIEME PARTIE"
       30 ...
          ...
     1000 LOAD "PROGRAMME 3"

etc...
```

SAUVEGARDE RAPIDE

L'instruction SAVE sauvant les lignes de programme, mais aussi les variables Basic, il peut être intéressant de taper directement, au clavier, l'instruction CLEAR avant d'effectuer la sauvegarde. CLEAR effaçant les variables et les tableaux, on gagne jusqu'à 50 % du temps de sauvegarde (surtout s'il existe des instructions comme DIM A(100,100) dans votre programme).

INSTRUCTION VAL

Il est possible d'utiliser la fonction VAL d'une chaîne de caractères contenant une expression complexe ou utilisant des variables Basic. On peut ainsi constituer des fonctions mathématiques entrées lors d'un INPUT pour les exécuter avec des variables déjà dans le programme. Pour obtenir les valeurs de 1 à 10 de la fonction Ax+B, il faut écrire le programme suivant :

```
10 LET A=10
20 LET B=5
30 FOR X=1 TO 10
40 PRINT A*X+B
50 NEXT X
```

Mais si vous désirez changer de fonction, il faut modifier la ligne 40. Avec l'instruction VAL, il est possible d'écrire ceci :

```
10 INPUT A
20 INPUT B
30 INPUT FONC$
40 FOR X=1 TO 10
50 PRINT VAL FONC$
60 NEXT X
```

LANGAGE MACHINE DANS UNE INSTRUCTION REM

C'est la solution la plus simple pour les programmes en L.M de petite taille. La première adresse de la routine sera 16514 puisque la mémoire vive utilisée par le Basic commence en 16509, le Basic prenant deux octets pour le numéro de ligne, deux octets pour sa longueur et un octet pour l'instruction REM. Vous devez placer, après l'instruction REM, autant de caractères qu'en comprend votre sous-programme en langage machine.

Exemple

```
10 REM 0123456789
20 POKE 16514,201
30 LET L=USR 16514
```

Remarque : l'instruction REM doit toujours être la première ligne du programme pour qu'elle ne soit pas déplacée par le Basic.

LANGAGE MACHINE DANS UNE VARIABLE

C'est un peu plus délicat, car votre programme doit être relocatable, c'est-à-dire ne comportant que des sauts relatifs, la variable étant déplacés dans la mémoire.

LANGAGE MACHINE EN HAUT DE MEMOIRE

Il suffit pour cela de modifier la variable système RAMTOP. Si vous avez besoin de 30 octets, vous devrez taper directement au clavier :

```
POKE 16388,226
POKE 16389,67
```

puis NEW

RAMTOP étant égal à 17407 (version 1Ko), 17407-30 = 226+256*67 (voir tableau de conversion en annexe).

Remarque : le sous-programme en langage machine ainsi établi n'est pas sauvé par l'instruction SAVE.

GAIN DE PLACE EN MEMOIRE CENTRALE

Si vous possédez une version 1Ko, il est nécessaire de savoir que :

- une ligne prend 5 caractères (2 pour le numéro de ligne, 2 pour le nombre d'octets et 1 pour le NEWLINE) ;
- un mot clé 1 caractère ;
- une variable 1 caractère dans le programme et 6 dans la mémoire ;
- un numérique 6 caractères.

Exemple

```
10 LET X = 1 (NEWLINE)
 ↑  ↑  ↑  ↑  ↑  ↑
 4  1  7  1  6  1  =  20 caractères
```

HORLOGE TEMPS REEL

La variable système FRAMES compte le nombre de trames affichées sur l'écran. Cette variable est donc décrémentée 50 fois par seconde et peut servir de compteur pour une horloge (précision du secteur).

Remarque : l'instruction PAUSE modifie la valeur des deux octets de FRAMES.

CARACTERES GEANTS

Le générateur de caractères du ZX-81 étant placé en mémoire morte, il est possible de le lire pour créer des caractères en grand format. L'adresse de cette table est située en 7680, on obtient l'adresse exact d'un caractère par la formule :

```
10 FOR I=0 TO 6
20 PRINT PEEK (7680+(8*CODE C$)+I)
30 NEXT I
```

où C$ correspond au caractère, et I aux 7 lignes qui le composent.

HAUTE-RESOLUTION SUR L'IMPRIMANTE

Pour obtenir des graphiques haute-résolution, sur l'imprimante, il suffit de recopier le programme de la mémoire morte du ZX-81 placé en 2161 en haut de mémoire, à partir de 31744 et en modifiant les adresses du tampon imprimante :

```
10 FOR I=0 TO 112
20 POKE 31744+I,PEEK (2161+I)
30 NEXT I
40 POKE 31800,63
50 POKE 31857,201
```

La représentation graphique de chaque caractère doit être placée dans la mémoire à partir de 32255. Un trait de 8 points correspond à 255, un point à droite à 1 et à gauche 128, etc.

Pour imprimer ce qui est contenu dans le tampon en 32255, il faut effectuer :

```
1000 FOR I=0 TO 31
1010 POKE 16444+I,I
1020 NEXT I
1030 LET L=USR 31744
```

Remarque : le programme n'affiche qu'une seule ligne à la fois et non 8 (matrice 8x8). Il faut réserver de la place mémoire en tapant directement au clavier : POKE 16389,125 puis NEW. Ce programme ne peut être mis en oeuvre que si vous possédez une extension 16Ko.

CONVERSION BINAIRE

Binaire	*Décimal*	*Hexadécimal*	*Octal*
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	10
1001	9	9	11
1010	10	A	12
1011	11	B	13
1100	12	C	14
1101	13	D	15
1110	14	E	16
1111	15	F	17

Exemple : D5E9 hexa. → 1101 0101 1110 1001

Tableau 1 : octet bas (de 0 à 255)

Dec.	*Hexa*	*Dec.*	*Hexa*	*Dec.*	*Hexa*	*Dec.*	*Hexa*
0	00	44	2C	88	58	132	84
1	01	45	2D	89	59	133	85
2	02	46	2E	90	5A	134	86
3	03	47	2F	91	5B	135	87
4	04	48	30	92	5C	136	88
5	05	49	31	93	5D	137	89
6	06	50	32	94	5E	138	8A
7	07	51	33	95	5F	139	8B
8	08	52	34	96	60	140	8C
9	09	53	35	97	61	141	8D
10	0A	54	36	98	62	142	8E
11	0B	55	37	99	63	143	8F
12	0C	56	38	100	64	144	90
13	0D	57	39	101	65	145	91
14	0E	58	3A	102	66	146	92
15	0F	59	3B	103	67	147	93
16	10	60	3C	104	68	148	94
17	11	61	3D	105	69	149	95
18	12	62	3E	106	6A	150	96
19	13	63	3F	107	6B	151	97
20	14	64	40	108	6C	152	98
21	15	65	41	109	6D	153	99
22	16	66	42	110	6E	154	9A
23	17	67	43	111	6F	155	9B
24	18	68	44	112	70	156	9C
25	19	69	45	113	71	157	9D
26	1A	70	46	114	72	158	9E
27	1B	71	47	115	73	159	9F
28	1C	72	48	116	74	160	A0
29	1D	73	49	117	75	161	A1
30	1E	74	4A	118	76	162	A2
31	1F	75	4B	119	77	163	A3
32	20	76	4C	120	78	164	A4
33	21	77	4D	121	79	165	A5
34	22	78	4E	122	7A	166	A6
35	23	79	4F	123	7B	167	A7
36	24	80	50	124	7C	168	A8
37	25	81	51	125	7D	169	A9
38	26	82	52	126	7E	170	AA
39	27	83	53	127	7F	171	AB
40	28	84	54	128	80	172	AC
41	29	85	55	129	81	173	AD
42	2A	86	56	130	82	174	AE
43	2B	87	57	131	83	175	AF

Dec.	Hexa	Dec.	Hexa	Dec.	Hexa	Dec.	Hexa
176	B0	196	C4	216	D8	236	EC
177	B1	197	C5	217	D9	237	ED
178	B2	198	C6	218	DA	238	EE
179	B3	199	C7	219	DB	239	EF
180	B4	200	C8	220	DC	240	F0
181	B5	201	C9	221	DD	241	F1
182	B6	202	CA	222	DE	242	F2
183	B7	203	CB	223	DF	243	F3
184	B8	204	CC	224	E0	244	F4
185	B9	205	CD	225	E1	245	F5
186	BA	206	CE	226	E2	246	F6
187	BB	207	CF	227	E3	247	F7
188	BC	208	D0	228	E4	248	F8
189	BD	209	D1	229	E5	249	F9
190	BE	210	D2	230	E6	250	FA
191	BF	211	D3	231	E7	251	FB
192	C0	212	D4	232	E8	252	FC
193	C1	213	D5	233	E9	253	FD
194	C2	214	D6	234	EA	254	FE
195	C3	215	D7	235	EB	255	FF

Tableau 2 : octet haut (de 0 à 65280)

Dec.	Hexa	Dec.	Hexa	Dec.	Hexa	Dec.	Hexa
0	00	4864	13	9728	26	14592	39
256	01	5120	14	9984	27	14848	3A
512	02	5376	15	10240	28	15104	3B
768	03	5632	16	10496	29	15360	3C
1024	04	5888	17	10752	2A	15616	3D
1280	05	6144	18	11008	2B	15872	3E
1536	06	6400	19	11264	2C	16128	3F
1792	07	6656	1A	11520	2D	16384	40
2048	08	6912	1B	11776	2E	16640	41
2304	09	7168	1C	12032	2F	16896	42
2560	0A	7424	1D	12288	30	17152	43
2816	0B	7680	1E	12544	31	17408	44
3072	0C	7936	1F	12800	32	17664	45
3328	0D	8192	20	13056	33	17920	46
3584	0E	8448	21	13312	34	18176	47
3840	0F	8704	22	13568	35	18432	48
4096	10	8960	23	13824	36	18688	49
4352	11	92&6	24	14080	37	18944	4A
4608	12	9472	25	14336	38	19200	4B

CONVERSION DECIMAL/HEXADECIMAL

Dec.	*Hexa*	*Dec*	*Hexa*	*Dec*	*Hexa*	*Dec.*	*Hexa*
19456	4C	30976	79	42496	A6	54016	D3
18172	4D	32232	7A	42752	A7	54272	D4
19968	4E	31488	7B	43008	A8	54528	D5
20224	4F	31744	7C	43264	A9	54784	D6
20480	50	32000	7D	43520	AA	55040	D7
20736	51	32256	7E	43776	AB	55296	D8
20992	52	32512	7F	44032	AC	55552	D9
32248	53	32768	80	44288	AD	55808	DA
21504	54	33024	81	44544	AE	56064	DB
21760	55	33280	82	44800	AF	56320	DC
22016	56	33536	83	45056	B0	56576	DD
22272	57	33792	84	45312	B1	56832	DE
22528	58	34048	85	45568	B2	57088	DF
22784	59	34304	86	45824	B3	57344	E0
23040	5A	34560	87	46080	B4	57600	E1
23296	5B	34816	88	46336	B5	57856	E2
23552	5C	35072	89	46592	B6	58112	E3
23808	5D	35328	8A	46848	B7	58368	E4
24064	5E	35584	8B	47104	B8	58624	E5
24320	5F	35840	8C	47360	B9	58880	E6
24576	60	36096	8D	47616	BA	59136	E7
24832	61	36352	8E	47872	BB	59392	E8
25088	62	36608	8F	48128	BC	59648	E9
25344	63	36864	90	48384	BD	59904	EA
25600	64	37120	91	48640	BE	60160	EB
25856	65	37376	92	48896	BF	60416	EC
26112	66	37632	93	49152	C0	60672	ED
26368	67	37888	94	49408	C1	60928	EE
26624	68	38144	95	49664	C2	61184	EF
26880	69	38400	96	49920	C3	61440	F0
27136	6A	38656	97	50176	C4	61696	F1
27392	6B	38912	98	50432	C5	61952	F2
27648	6C	39168	99	50688	C6	62208	F3
27904	6D	39424	9A	50944	C7	62464	F4
28160	6E	39680	9B	51200	C8	62720	F5
28416	6F	39936	9C	51456	C9	62976	F6
28672	70	40192	9D	51712	CA	63232	F7
28928	71	40448	9E	51968	CB	63488	F8
29184	72	40704	9F	52224	CC	63744	F9
29440	73	40960	A0	52480	CD	64000	FA
29696	74	41216	A1	52736	CE	64256	FB
29952	75	41472	A2	52992	CF	64512	FC
30208	76	41728	A3	53248	D0	64768	FD
30464	77	41984	A4	53504	D1	65024	FE
30720	78	42240	A5	53760	D2	65280	FF

Exemple : 1C9E hexa.

1C (haut) → 7168
9E (bas) → +158
} = 7326 décimal

INDEX

	Pages
ABS	9, 72
ACS	9, 73
Addition	70
AND	14, 73
ASN	9, 73
ATN	9, 73
AT	13
BERG	73, 77
Binaire	87
BLACK	16, 74
BOX	16, 74
BRBUFF	78
Carte haute-résolution	16, 74
Carte génératrice de caractères	17
Carte son	63, 79
CDFLAG	78
CH-ADD	66, 76
CHR$	10, 73
Circuits intégrés	60
CLEAR	11, 69, 82
CLEAR H. R.	16, 74
CLR	17
CLS	11, 67
CODE	10, 73
Codes clavier	27, 66
Codes écran	23, 66, 85
Codes erreurs	19, 66
Codes des mots clefs	15, 67, 70
Codes Z80	40
Connecteurs	59
CONT	11, 68
Constantes	70
COORDS	74, 78
COPY	11, 67
COPY H.R.	16, 74
COS	9, 73

Décimal 88
Désassemblage 55
DEST 76
D-FILE 75
DF-CC 66, 75
DF-SZ 77
DIM 11, 69
Division 70
DOWN 16, 74
DRAW 16, 74
Ecran 29, 66
EDIT 11
E-LINE 67, 69, 76
E-PPC 75
ERR-NR 75
ERR-SP 75
EXP 9, 73
FAST 11, 68
FLAGS 69, 75
FLAGX 77
FOR 11, 68
FRAMES 77
GOSUB 11, 68
GOTO 11, 68
Hexadécimal 88
IF 11, 68
Imprimante 85
Initialisation 66
INKEY$ 12
INPUT 12, 68
INT 9, 73
Instructions Z80 34
Langage machine 31
LAST-K 77
LEFT 16, 74
LEN 10, 73
LET 12, 69
LIST 12, 67
LLIST 12, 67
LN 9, 73
LOAD 12, 66, 81
LPRINT 12, 67
MARGIN 77
MEM 72, 77
MEMBOT 78
Mémoires 65
MODE 75

MOVE 16, 74
Multiplication 70
NEW 12, 66
NEXT 12, 68
NOT 14, 73
NXTLIN 77
OLDPPC 77
OR 14, 73
PAUSE 12, 69
PEEK 10, 73
PI 9, 70
PLOT 12, 67
PLOT H.R. 16, 74
POKE 10, 68
PPC 75
PR-CC 78
PRINT 12, 67, 70, 81
PRINT A$ (H.R.) 16, 74
PRINT AT 13, 67
PRINT TAB 13
QCOPY 17
QLOAD 17
QPRNT 17
RAMTOP 65, 75
RAND 13, 68
REBOND 77
Registres Z80 32
REM 13, 68, 83
RETURN 13, 68
RIGHT 16, 74
RND 13
Rom Basic 66
RST (routines) 66
RUN 13, 68
RUBOUT 13
SAVE 13, 66, 81
Schéma ZX-81 33
SCROLL 13, 67
SCROLL H.R. 16, 74
SEED 77
SGN 9, 72
SIN 9, 73
SINC 17
SLOW 13, 66, 68
S-POSN 78
SQR 9, 73
STEP 11

STKEND 69, 73, 76
STKBOT 66, 76
STOP 13, 68
S-TOP 77
STR$ 10, 70, 73
STRLEN 77
TAB 13
T-ADDR 77
TAN 9, 73
THEN 11
TO 10
UNPLOT 13, 67
UP 16, 74
USR 10, 73
VAL 10, 73, 83
Variables basic 84
Variables système 75
VARS 76
VERSN 75
Virgule flottante 66, 70
WHITE 16, 74
X-PTR 76

Achevé d'imprimer en septembre 1983
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : septembre 1983

N° d'impression : 307015
N° d'édition : 86595-86-1
ISBN : 2-86595-086-7

mémento

Comment gagner du temps... Sans en perdre à glaner de ci de là tous les renseignements techniques dont vous avez besoin pour bien utiliser votre Sinclair. Les **"Clefs"** c'est : ● La liste des instructions Basic commentées ● les mnémoniques de l'assembleur Z 80 et leurs codes objets ● les points d'entrée de la ROM Basic ● des explications sur les variables système. Et c'est également : ● les caractéristiques des principales extensions ● une liste d'astuces pour mieux utiliser l'ecran, les cassettes et les programmes en langage machine.

CLEFS POUR LE ZX 81

Editions du P.S.I.
B.P. 86
77402 Lagny/Marne
France

ISBN 2.86595.086.7

82,00 FF

ouverture Mariane de Nayer

Imprimé en France ✶ ✶

CLEFS POUR LE ZX 81

Jean-François Séhan